



TESIS

Tesis - TF185471

STUDI SISTEM MONITORING PREDIKSI KEANDALAN *REAL-TIME* PADA SISTEM PENGENDALIAN KECEPATAN GENERATOR TURBIN ANGIN DENGAN KESALAHAN SENSOR

LILIK AYURANI
NRP. 02311650012002

DOSEN PEMBIMBING
Dr. Katherin Indriawati, S.T., M.T

PROGRAM MAGISTER
BIDANG KEAHLIAN REKAYASA INSTRUMENTASI INDUSTRI
JURUSAN TEKNIK FISIKA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2019



TESIS

Tesis - TF185471

STUDI SISTEM MONITORING PREDIKSI KEANDALAN *REAL-TIME* PADA SISTEM PENGENDALIAN KECEPATAN GENERATOR TURBIN ANGIN DENGAN KESALAHAN SENSOR

LILIK AYURANI
NRP. 02311650012002

DOSEN PEMBIMBING
Dr. Katherin Indriawati, S.T., M.T

PROGRAM MAGISTER
BIDANG KEAHLIAN REKAYASA INSTRUMENTASI INDUSTRI
JURUSAN TEKNIK FISIKA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2019

Halaman ini sengaja dikosongkan

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Teknik (M.T)
di
Institut Teknologi Sepuluh Nopember

Oleh :

Lilik Ayurani

NRP. 02311650012002

Tanggal Ujian: 9 Januari 2019


Periode Wisuda: Maret 2019

Disetujui oleh :


1. Dr. Katherin Indriawati, S.T., M.T
NIP. 19760523 200012 2 001

 (Pembimbing)

2. Dr. Ir. Ali Musyafa', M. Sc
NIP. 19600901 198701 1 001


 (Penguji)

3. Dr. Imam Abadi, S. ., M.T
NIP. 19761006 199903 1 002

 (Penguji)



Dekan Fakultas Teknologi Industri,


Dr. Bambang Lelono Widjiantoro, S.T., M.T.

NIP: 19690507 199512 1 001

Halaman ini sengaja dikosongkan

STUDI SISTEM MONITORING PREDIKSI KEANDALAN *REAL-TIME* PADA SISTEM PENGENDALIAN KECEPATAN GENERATOR TURBIN ANGIN DENGAN KESALAHAN SENSOR

Nama Mahasiswa : Lilik Ayurani
NRP : 02311650012002
Pembimbing : Dr. Katherin Indriawati, S.T., M.T

ABSTRAK

Peningkatan jumlah penduduk seiring dengan peningkatan kebutuhan energi listrik yang ada di Indonesia. Salah satu cara yang sedang digalakkan untuk pemenuhan kebutuhan listrik melalui energi alternatif adalah dengan menggunakan turbin angin. Sementara sistem turbin angin sedang digunakan, keandalannya akan menurun secara bertahap. Penelitian ini bertujuan untuk menentukan parameter desain yang mempengaruhi prediksi keandalan *real-time* pada sistem pengendalian kecepatan generator turbin angin. Tiga langkah yang perlu dilakukan yaitu perancangan observer, perancangan algoritma prediksi kesalahan sensor dan perancangan algoritma prediksi keandalan. Perancangan *observer* digunakan untuk mengestimasi kesalahan sensor dari variabel yang terukur, kemudian hasil estimasi digunakan untuk menghitung prediksi kesalahan sensor melalui algoritma *exponential smoothing*. Hasil dari prediksi kesalahan sensor ini secara langsung digunakan untuk prediksi keandalan *real-time*. Waktu kegagalan riil ketika kecepatan generator lebih besar dari 1.2 pu dan kecepatan generator dibawah 0.5 pu menandakan keandalan mulai turun. Variasi diberikan terhadap *time interval* sebesar 1 detik, 5 detik, 10 detik dan 50 detik, sedangkan setiap *time interval* diberikan variasi jumlah prediksi sebesar 1, 5 dan 10. Hasil simulasi menunjukkan bahwa prediksi keandalan dengan *time interval* sebesar 1 detik lebih tepat jika dibandingkan variasi *time interval* sebesar 5 detik, 10 detik, dan 50 detik. Jumlah prediksi mempengaruhi ketelitian prediksi keandalan *real-time* yang dihasilkan, semakin besar jumlah prediksi maka semakin teliti. Dengan *time interval* 1 detik, jumlah prediksi sebesar 10 menghasilkan prediksi keandalan *real-time* lebih tepat dibandingkan jumlah prediksi sebesar 1 dan 5.

Kata kunci : Turbin angin, *observer*, prediksi kesalahan sensor, prediksi keandalan *real-time*.

Halaman ini sengaja dikosongkan

STUDY OF REAL-TIME RELIABILITY PREDICTION MONITORING SYSTEM FOR WIND TURBINES GENERATOR SPEED CONTROL SYSTEMS WITH SENSOR FAULT

Nama Mahasiswa : Lilik Ayurani
NRP : 02311650012002
Pembimbing : Dr. Katherin Indriawati, S.T., M.T

ABSTRACT

Indonesia is known as the largest archipelagic country in the world. It has a significant growth in its population every year. As the result, the energy demand in Indonesia continues to develop. One of the method of helping meet energy needs is by introducing wind turbine as alternative energy source. However, the reliability on the turbine system will be decreasing steadily if it is being continually exploited. This study aims to determine the design parameters that is affecting the real-time reliability prediction for wind turbine generator speed control systems. There are three steps that need to be done, namely observer design, sensor fault prediction algorithm design, and reliability prediction algorithm design. Observer design was used to estimate sensor fault from the measured variable and the estimated result was used to calculate sensor fault prediction using exponential smoothing algorithm. The calculated result will then be directly exploited to predict real-time reliability. We obtained that the failure time correspond with a decrease in the reliability when the generator speed is greater than 1.2 pu and below 0.5 pu. Time interval was varied by 1s, 5s, 10s, and 50s. In each of the time interval, the variation number of prediction given was of 1, 5, and 10. Simulation result has shown that reliability prediction with time interval 1s was more precise compared to time interval variation of 5s, 10s, and 50s. The number of prediction affects real-time reliability prediction fidelity resulted. With the time interval of 1s, the number of prediction of 10 results in accurate real-time reliability prediction compared to number of prediction of 1 and 5.

Key words : Wind turbine generator, observer, sensor fault prediction, real-time reliability prediction.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdulillah, segala puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat, hidayat, serta petunjuk-Nya sehingga penulis dapat menyelesaikan tesis pada jenjang magister yang berjudul “*Studi Sistem Monitoring Prediksi Keandalan Real-Time pada Sistem Pengendalian Kecepatan Generator Turbin Angin Dengan Kesalahan Sensor*”. Penulis telah banyak mendapatkan bantuan dari berbagai pihak dalam menyelesaikan tugas akhir ini. Atas berbagai bantuan itu penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Orang tua tercinta beserta saudara yang mendukung dalam pekerjaan tesis ini.
2. Ibu Dr. Katherin Indriawati, S.T., M.T selaku pembimbing yang telah sabar memberikan petunjuk, bimbingan, serta ilmu yang sangat bermanfaat dalam terselesainya tesis ini.
3. Bapak Dr. Ir. Ali Musyafa', M. Sc. Dan Bapak Dr. Imam Abadi, S.T., M.T selaku dosen penguji yang memberikan bimbingan dan saran-saran untuk perbaikan tesis ini.
4. Prof., Dr., Ir., Sekartedjo, M. Sc selaku dosen wali yang telah memberikan motivasi, bimbingan, dan ilmunya yang sangat bermanfaat.
5. Bapak dan Ibu dosen Teknik Fisika yang telah memberikan ilmu selama kuliah yang sudah banyak membantu penulis selama kuliah.
6. Teman-teman Teknik Fisika angkatan 2012 yang memberikan dukungan dan semangat kepada penulis.
7. Teman-teman seperjuangan kuliah S2 dan pejuang Tesis yang telah berbagi ilmu dan pengalaman.
8. Semua pihak yang telah mendukung dan memberikan doa yang tidak bisa penulis sebutkan satu per satu.

Penulis menyadari bahwa penulisan laporan tesis ini tidaklah sempurna. Oleh karena itu, sangat diharapkan kritik dan saran yang membangun dari semua pihak sehingga mencapai sesuatu yang lebih baik lagi. Penulis juga berharap semoga laporan ini dapat menambah wawasan yang bermanfaat bagi pembaca.

Surabaya, Januari 2019

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	Error! Bookmark not defined.
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
DAFTAR NOTASI	xvii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	4
1.3. Tujuan Penelitian	5
1.4. Batasan Masalah	5
1.5. Manfaat Penelitian	5
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI.....	7
2.1. Turbin Angin.....	7
2.2. Komponen Turbin Angin.....	9
2.3. Pemodelan Dinamik dari Sistem Generator Turbin Angin.....	9
2.4. Sistem Kendali PI	12
2.5. Kesalahan Sistem Kontrol.....	13
2.6. Karakteristik Sistem Pengukuran.....	15
2.7. Tachometer	16
2.8. <i>Observer</i>	16
2.9. Algoritma <i>Exponential Smoothing</i> untuk Prediksi Kesalahan.....	19
2.10. Keandalan	21
2.11. Prediksi Keandalan <i>Real-Time</i>	21

BAB 3 METODA PENELITIAN	25
3.1 Studi Literatur	26
3.2 Pemodelan Sistem Kontrol Generator Turbin Angin.....	26
3.3 Perancangan Algoritma <i>Observer</i>	31
3.4 Pengujian Algoritma <i>Observer</i>	36
3.5 Perancangan Algoritma Prediksi Kesalahan	37
3.6 Pengujian Algoritma Prediksi Kesalahan Sensor.....	38
3.7 Perancangan Algoritma Prediksi Keandalan <i>Real-Time</i>	38
3.8 Pengujian Algoritma Prediksi Keandalan <i>Real-time</i>	39
BAB 4 HASIL DAN PEMBAHASAN	41
4.1. Hasil Uji <i>Observer</i>	41
4.2. Hasil Uji Algoritma Prediksi Kesalahan Sensor	41
4.3. Hasil Uji Algoritma Prediksi Keandalan <i>Real-time</i>	46
B AB 5 KESIMPULAN	53
5.1. Kesimpulan	53
5.2. Saran.....	53
DAFTAR PUSTAKA.....	55
LAMPIRAN A	A-1
LAMPIRAN B.....	B-1
LAMPIRAN C.....	C-1

DAFTAR GAMBAR

Gambar 2.1 Klasifikasi turbin angin berdasarkan arah putarnya (Jha, 2011)	8
Gambar 2.2 Komponen Turbin Angin (Jha, 2011).....	9
Gambar 2.3 Blok diagram pemodelan turbin angin (Sloth, 2011)	10
Gambar 2.4 Variasi dari C_p dari Tip Speed Ratio (Shaker, 2012).....	12
Gambar 2.5 Diagram blok sistem pengendalian tertutup (Basilio, 2002)	13
Gambar 2.6 Jenis kesalahan pada sensor (Bruno, Oussama, & Frans, 2003)	14
Gambar 2.7 Skema Tachometer DC (Brodgesell, 2003).....	16
Gambar 2.8 Observer (Ogata, 2004)	19
Gambar 2.9 Kestabilan pole placement sistem diskrit (M. Chen, 2016).....	19
Gambar 3.1 Diagram Alir Penelitian.....	25
Gambar 3.2 Diagram Blok Sistem Pengendalian Kecepatan Turbin Angin	31
Gambar 3.3 Diagram blok sistem pengendalian yang telah terintegrasi dengan observer	36
Gambar 3.4 Diagram blok sistem prediksi keandalan real-time	39
Gambar 4.1 Grafik hasil estimasi kesalahan sensor	41
Gambar 4.2 Grafik hasil uji prediksi kesalahan sensor dengan time interval 1 detik	42
Gambar 4.3 Grafik hasil uji prediksi kesalahan sensor dengan time interval 5 detik	43
Gambar 4.4 Grafik hasil uji prediksi kesalahan sensor dengan time interval 10 detik	44
Gambar 4.5 Grafik hasil uji prediksi kesalahan sensor dengan time interval 50 detik	45
Gambar 4.6 Grafik hasil uji prediksi keandalan <i>real-time</i> dengan <i>time interval</i> 1 detik	47
Gambar 4.7 Grafik hasil uji prediksi keandalan <i>real-time</i> dengan <i>time interval</i> 5 detik	48
Gambar 4.8 Grafik hasil uji prediksi keandalan <i>real-time</i> dengan <i>time interval</i> 10 detik	49

Gambar 4.9 Grafik hasil uji prediksi keandalan *real-time* dengan *time interval* 50 detik..... 50

DAFTAR TABEL

Tabel 2.1 Klasifikasi Turbin Angin Berdasarkan Kapasitas Energi Listrik yang Dihasilkan (Jha, 2011)	7
Tabel 3.1 Spesifikasi Turbin Angin Pada Simulasi (J. F. Manwell & J. G. McGowan, 2009)	26
Tabel 3.2 Parameter – parameter pada pemodelan turbin angin (Nicholas, dkk. 2003)	27
Tabel 4.1 Hasil Uji Prediksi Keandalan Real-Time	51

Halaman ini sengaja dikosongkan

DAFTAR NOTASI

$\hat{x}(k)$: estimasi <i>state</i> $x(k)$
$\hat{x}(k + 1)$: estimasi <i>state</i> $x(k+1)$
$\hat{y}(k)$: estimasi $y(k)$
$\hat{f}(k)$: estimasi kesalahan sensor $f(k)$
$\hat{f}(k + 1)$: prediksi kesalahan sensor satu step
$\hat{f}(k + 2)$: prediksi kesalahan sensor dua step
$\hat{f}(k + 3)$: prediksi kesalahan sensor tiga step
$\hat{f}(k + 4)$: prediksi kesalahan sensor empat step
$\hat{f}(k + 5)$: prediksi kesalahan sensor lima step
A	: matriks keadaan (<i>state</i>)
b	: koefisien gesekan yang merugikan rotor
B	: matriks kendali
C	: matriks keluaran
l	: step prediksi untuk prediksi kesalahan sensor
n	: <i>discrete time index</i>
r	: step prediksi untuk prediksi keandalan <i>real-time</i>
s_n	: faktor pertumbuhan linear yang diestimasi
u_n	: hasil estimasi yang diperhalus
$x(k)$: vektor keadaan (<i>state</i>)
$x(k+1)$: turunan vektor keadaan
$y(k)$: vektor keluaran

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Indonesia merupakan negara kepulauan terbesar di dunia dan pertumbuhan penduduk Indonesia pun terus meningkat setiap tahun. Peningkatan jumlah penduduk seiring dengan peningkatan kebutuhan energi listrik yang ada di Indonesia. Kebutuhan listrik menjadi permasalahan yang sangat penting di zaman modern seperti ini. Maka dari itu banyak langkah-langkah yang dilakukan oleh pemerintah guna mengatasi tingginya kebutuhan listrik. Salah satunya dengan memanfaatkan sumber energi alternatif yang dikonversikan menjadi energi listrik seperti biomassa, *photovoltaic* dan turbin angin dan lain sebagainya.

Salah satu cara yang sedang digalakkan untuk pemenuhan kebutuhan listrik melalui energi alternatif adalah dengan menggunakan turbin angin. Turbin angin adalah suatu alat yang digunakan untuk mengubah energi angin menjadi listrik. Sebagai generator listrik, turbin angin di hubungkan ke beberapa jaringan listrik, jaringan ini meliputi rangkaian pengisian baterai, daya hunian residensial sistem, atau jaringan terisolasi. Dalam sistem turbin angin, proses konversi menggunakan gaya aerodinamik pada gaya (*lift*) angkat untuk menghasilkan torsi positif pada putaran poros, putaran pada *blades* yang di hubungkan pada generator menjadikan generator ikut berputar, dimana dalam hal ini terjadinya proses konversi energi mekanik menjadi energi listrik (Manwell & Mc Gowan, 2009).

Penelitian mengenai kegagalan dan mekanisme kegagalan pada sistem generator turbin angin disebabkan oleh perubahan suhu pada generator, dimana efisiensi pada generator akan menurun. Dengan meningkatkan sistem pendingin akan mengurangi kegagalan dan dapat meningkatkan keandalan pada sistem generator turbin angin. Untuk penelitian lebih lanjut dalam meningkatkan keandalan disarankan untuk mengganti potongan stator magnetik dengan non-magnetik, dan penggunaan *brushless doubly-fed induction generators* (BDFIG) karena kegagalan pada rotor dan *collector rings* menyebabkan kurang lebih 7% terjadi kegagalan pada semua generator turbin angin (Shipurkar et al., 2015).

Sistem turbin angin membutuhkan tingkat keandalan dan keberlanjutan yang tinggi. Biaya pemeliharaan sistem turbin angin untuk menjamin keamanan diketahui cukup mahal (Charles-de-Gaulle-Str, 2010). Sehubungan dengan peningkatan kebutuhan listrik negara, upaya untuk mempertahankan ekstraksi energi angin harus dilakukan. Salah satu upaya yang diperlukan adalah peningkatan keandalan sistem turbin angin, baik melalui *preventive maintenance* maupun *predictive maintenance*. Perawatan prediktif suatu sistem telah banyak dikaji oleh berbagai kalangan, mulai dari para akademis maupun *engineer* seperti (Grall et al., 2002)(Usher et al., 1998)(Márquez & Pedregal, 2007). Salah satu teknik dalam perawatan prediktif dapat dilakukan dengan memonitor kondisi dan memprediksi sisa *life time* dari sistem tersebut, seperti yang diajukan oleh (Jardine et al., 2006). Dalam hal ini indikasi pada komponen dan sistem menunjukkan produkivitas sisa *life time*.

Keandalan sistem ditentukan saat sistem itu mulai beroperasi. Namun, keandalan sistem akan menurun selama penggunaannya karena degradasi komponen, atau gangguan dari luar. Kemungkinan sistem ini akan gagal yang mengakibatkan keandalan sistemnya semakin rendah. Oleh karena itu, keandalan sistem berkaitan dengan kinerja dari sistem, dan bertindak sebagai indikator alami kinerja sistem. Skema yang menghubungkan keandalan dengan indikasi umur sebuah sistem adalah evaluasi dan prediksi keandalan secara *real-time* (Xu et al., 2009). Penelitian dibidang ini telah berkembang sejak awal 1990-an di antaranya adalah (Kolarik & Lu, 2001)(Chinnam & Mohan, 2002)(Xu et al., 2008).

Monitoring keandalan merupakan usaha mengevaluasi kondisi sistem tersebut secara periodik sehingga sistem dapat melaksanakan fungsinya dengan baik, tanpa adanya kegagalan pada periode operasi yang ditentukan. Monitoring ini juga bisa digunakan untuk memprediksi keandalan sebuah sistem. Prediksi keandalan digunakan untuk menjejaki peningkatan keandalan dan dapat juga membantu mengevaluasi signifikansi kegagalan yang dilaporkan (Xu et al., 2008). Prediksi keandalan ini juga perlu diterapkan untuk sistem generator pada turbin angin, sehingga kegagalan yang akan terjadi pada turbin angin dapat dideteksi lebih dini. Dengan dilakukannya prediksi keandalan dapat meningkatkan umur

peralatan, mengurangi waktu *downtime* dan kerusakan yang lebih parah dapat dihindari.

Evaluasi dan prediksi keandalan secara *real-time* secara kasar dapat dibagi menjadi dua yaitu analisis regresi dan analisis deret waktu. Metode analisis regresi sering mengadopsi fungsi jalur degradasi yang dapat menggambarkan variasi kasar yang sesuai dengan variabel performansi selama siklus sistem beroperasi secara keseluruhan dan pengukuran *online* pada variabel performansi digunakan untuk mengidentifikasi fungsi jalur degradasi. Taylor *et al.* (2007) menggunakan metode analisis regresi untuk memodelkan variasi dari data performansi, dan keandalan kondisional diprediksi secara *real-time* berdasarkan pemilihan model (*model selection*). Chinnam & Mohan (2002) mempelajari estimasi keadaan *online* untuk komponen individual menggunakan sinyal degradasi. Variasi variabel performansi dimodelkan dengan model regresi polynomial secara umum dan parameter dalam model diestimasi menggunakan metode *least squares*.

Pada tahun (2006) Gebraeel mengajukan metode prediksi distribusi *residual life* untuk produk dengan jalur degradasi eksponensial. Pendekatan Bayesian diaplikasikan untuk mengupdate parameter dalam model jalur degradasi berdasarkan informasi *real-time* hasil monitoring kondisi. Termotivasi oleh Gebraeel (2006), Xu & Zhou (2006) mempelajari masalah prediksi keandalan secara *real-time* untuk produk dengan jalur degradasi nonlinear secara umum. Parameter dalam model degradasi diestimasi menggunakan pendekatan Bayesian dan keandalan prediktif dihitung melalui strategi Monte Carlo. Metode analisis deret waktu mengambil pengukuran *online* pada variabel performansi sebagai deret waktu, dan analisis deret waktu dengan beberapa pendekatan statistik atau kecerdasan buatan. Kolarik & Lu (2001) menganalisis sinyal degradasi produk individual melalui metode *exponential smoothing*. Chinnam (1999) menghitung keandalan prediktif produk berdasarkan asumsi distribusi normal.

Lu *et al.* (2001) mengadopsi model *state space* untuk mendeskripsikan data degradasi performansi multivarian dan mengaplikasikan Kalman *filtering* untuk memprediksi *mean* dan *covariance* pada pengukuran performansi. Lu *et al.* menggunakan pengukuran performansi yang diprediksi untuk menghitung

keandalan prediktif. Selain analisis regresi dan analisis deret waktu, *stochastic filtering* juga digunakan untuk prediksi keandalan secara *real-time*. Dalam (Pulkkinen et al., 2006) proses degradasi dimodelkan seperti rantai Markov *discrete time*. Lebih lanjut kombinasi dari monitoring proses degradasi dan variasi lingkungan ditinjau dari (Jeffrey P Kharoufeh & Cox, 2005) berdasarkan hasil dalam (Je P Kharoufeh, 2003). Dibawah asumsi lingkungan Markovian, dua jenis data sensor digunakan untuk mengestimasi distribusi *lifetime*, observasi lingkungan dan ukuran degradasi.

Keseluruhan itu *single* komponen, jadi masih sedikit yang digunakan untuk sistem dinamik yang terdiri atas beberapa komponen. Salah satu diantaranya adalah penelitian yang dilakukan oleh (Xu et al., 2009), mereka melakukan prediksi keandalan berbasis prediksi kesalahan *online* untuk sistem dinamis. Keandalan sistem diprediksi sesuai hasil prediksi kesalahan dengan menggunakan strategi simulasi Monte Carlo. Namun penelitian tersebut hanya membahas kesalahan yang bertindak sebagai input aditif dari sistem dinamik (persamaan variabel keadaan) sehingga tidak mampu memprediksi keandalan untuk kesalahan aditif pada *output* sistem. Diusulkan algoritma prediksi keandalan pada sistem pengendalian kecepatan generator turbin angin yang terdiri atas komponen turbin, *drive train*, generator dan *converter*. Kesalahan yang diteliti meliputi kesalahan aditif pada *output* sistem yaitu kesalahan sensor kecepatan generator.

1.2. Perumusan Masalah

Perumusan masalah dalam penelitian ini antara lain :

1. Bagaimana menggabungkan algoritma *observer* dan algoritma *exponential smoothing* pada sistem pengendalian kecepatan generator turbin angin?
2. Parameter desain apakah yang mempengaruhi prediksi keandalan *real-time* pada sistem pengendalian kecepatan generator turbin angin?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini berdasarkan rumusan masalah adalah sebagai berikut:

1. Mengetahui cara menggabungkan algoritma *observer* dan algoritma *exponential smoothing* sensor pada sistem pengendalian kecepatan turbin angin secara *real-time*.
2. Menentukan nilai parameter desain yang mempengaruhi prediksi keandalan *real-time* pada sistem pengendalian kecepatan generator turbin angin.

1.4. Batasan Masalah

Batasan masalah dalam penelitian ini diantaranya sebagai berikut:

1. Variabel yang diukur adalah kecepatan generator dengan kesalahan pada sensor.
2. Algoritma pengendalian yang digunakan pada penelitian ini yaitu jenis pengendalian PI.
3. Jenis turbin angin yang diteliti adalah jenis horizontal (*horizontal wind turbine* / HWT) dengan daya nominal 400 Watt.
4. Prediksi keandalan yang dimaksud pada penelitian ini adalah prediksi keandalan *real-time*, bukan prediksi keandalan yang diperoleh dari data *maintenance*.

1.5. Manfaat Penelitian

Penelitian prediksi keandalan untuk turbin angin memiliki prospek kedepan yang bermanfaat baik dalam segi ilmu maupun aplikasi. Penelitian ini diharapkan dapat menjadi acuan untuk meningkatkan umur peralatan, mengurangi waktu *downtime* dan kerusakan yang lebih parah dapat dihindari pada saat turbin angin beroperasi.

Halaman ini sengaja dikosongkan

BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI

2.1. Turbin Angin

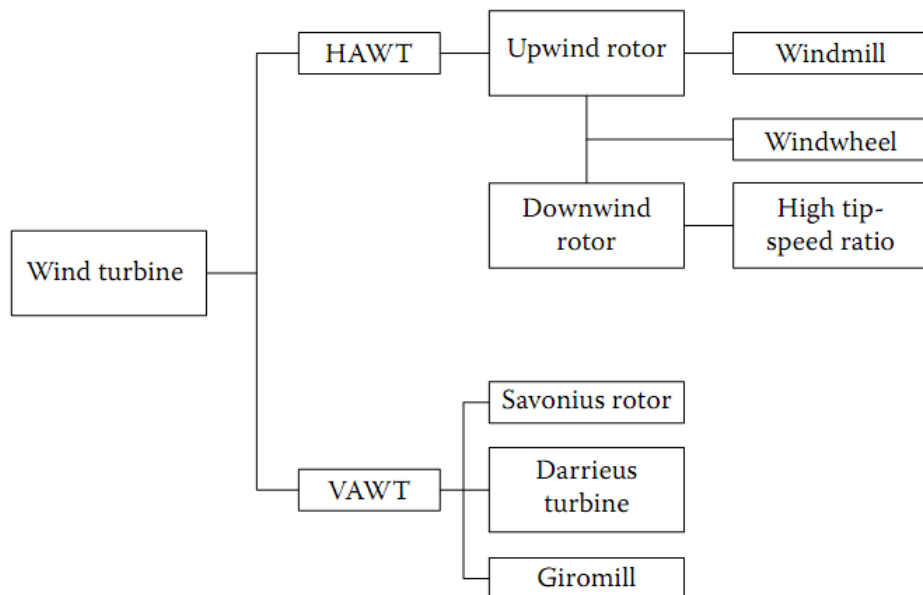
Turbin angin adalah suatu alat yang digunakan untuk mengubah energi angin menjadi listrik. Sebagai generator listrik, turbin angin dihubungkan ke beberapa jaringan listrik, jaringan ini meliputi rangkaian pengisian baterai, daya hunian residensial sistem, atau jaringan terisolasi. Dalam turbin angin modern, proses konversi menggunakan gaya aerodinamik pada gaya (*lift*) angkat untuk menghasilkan torsi positif pada putaran poros, putaran pada *blades* yang di *couple* pada generator menjadikan generator ikut berputar, dimana dalam hal ini terjadinya proses konversi energi mekanik menjadi energi listrik. Turbin angin menghasilkan daya tergantung dari interaksi antara rotor dan angin. Angin sendiri merupakan kombinasi dari angin rata-rata dan fluktuasi turbulensi dari aliran angin tersebut. Berdasarkan (Manwell & Mc Gowan, 2009) menunjukkan bahwa aspek utama dari kinerja turbin angin (berarti *output* daya dan beban rata-rata) ditentukan oleh gaya aerodinamik yang dihasilkan oleh angin rata-rata. Gaya periodik aerodinamik dapat disebabkan oleh angin geser, angin *off-axis*, dan rotasi rotor.

Klasifikasi turbin angin berdasarkan kapasitas energi listrik yang dihasilkan dibagi menjadi tiga kategori, yaitu turbin angin skala utilitas, skala kecil, dan skala menengah. Perbandingan masing-masing kategori turbin angin ditampilkan pada Tabel 2.1.

Tabel 2.1 Klasifikasi Turbin Angin Berdasarkan Kapasitas Energi Listrik yang Dihasilkan (Jha, 2011)

Kategori Turbin Angin	Kapasitas energi listrik yang dihasilkan
Skala utilitas	1,5 - 3,6 MW
Skala kecil	≥ 20 kW
Skala menengah	20 – 100 kW

Turbin angin juga dapat digolongkan berdasarkan sumbu putarnya, yaitu turbin angin sumbu horizontal dan turbin angin sumbu vertikal. Secara umum, penggolongan turbin angin berdasarkan sumbu putarnya ditampilkan pada Gambar 2.1.



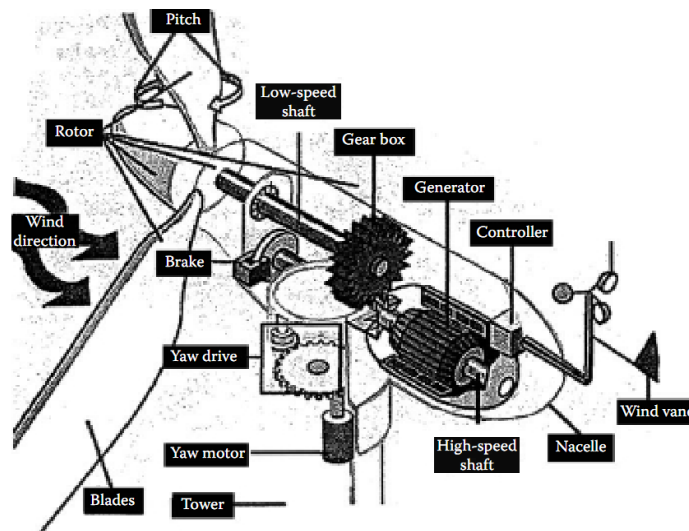
Gambar 2.1 Klasifikasi turbin angin berdasarkan arah putarnya (Jha, 2011)

Turbin angin tipe VAWT memiliki bentuk lebih sederhana dari pada sistem HAWT. Tipe turbin angin HAWT dapat digunakan pada daerah yang memiliki kecepatan angin rendah maupun kecepatan angin tinggi. Turbin angin tipe HAWT sangat variatif terkait dengan konfigurasi, jumlah *blade*, ukuran rotor, tipe *blade*, dan sistem kontrol yang sangat beragam. Perkembangan ukuran dan tipe turbin angin juga sangat variatif. Turbin angin yang memiliki dimensi dan tipe *blade* yang berbeda akan menghasilkan kinerja yang berbeda pula, terutama efisiensi yang dihasilkan. Turbin angin yang memiliki jumlah *blade* dua atau tiga sering digunakan untuk tujuan pembangkit tenaga listrik, karena turbin angin ini meskipun mempunyai torsi yang relatif kurang besar namun memiliki putaran rotor tinggi yang dapat ditunjukkan melalui hubungan nilai efisiensi (η), *tip speed ration* (λ), dan torsi (CQ). Sedangkan turbin angin yang memiliki jumlah *blade* lebih banyak dapat menghasilkan torsi yang lebih besar, namun demikian turbin belum tentu mampu menghasilkan efisiensi yang tinggi. Karena pada daerah

dengan kecepatan angin yang tidak terlalu tinggi diperlukan mampu menghasilkan jumlah putaran turbin yang diharapkan(J.H. Laks,2009).

2.2. Komponen Turbin Angin

Turbin angin yang tersusun atas komponen yang ditunjukkan oleh Gambar 2.2. Turbin angin terdiri dari komponen utama: bilah, *gearbox*, generator dan kendali. Bilah turbin angin adalah komponen utama turbin angin untuk menangkap energi angin, energi tersebut dikonversi menjadi energi mekanik dan listrik.



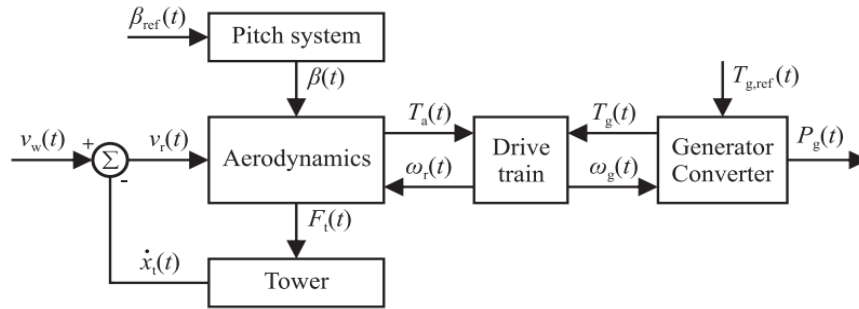
Gambar 2.2 Komponen Turbin Angin (Jha, 2011)

Gambar 2.2 menunjukkan komponen-komponen turbin angin secara lengkap. Komponen-komponen pada turbin angin antara lain: rotor, bilah (*blade*), pengereman (*brake*), poros kecepatan rendah (*low-speed shaft*), *gearbox*, generator, poros kecepatan tinggi (*high-speed shaft*), *nacelle*, dan tower. Sistem pengendalian yang digunakan pada turbin angin antara lain pengendalian sudut *pitch*, pengendalian sudut *yaw*, serta pengendalian untuk sistem pengereman.

2.3. Pemodelan Dinamik dari Sistem Generator Turbin Angin

Suatu sistem disusun dari bermacam-macam komponen yang saling berhubungan dalam menjalankan suatu aksi untuk mencapai tujuan yang

diinginkan. Suatu prasyarat dasar pada kebanyakan strategi yang dipergunakan dalam membangun suatu sistem kendali adalah kemampuan dalam memodelkan secara matematika dari sistem yang akan dikendalikan. Model matematika dari suatu sistem dapat berupa persamaan diferensial, fungsi alih atau persamaan ruang keadaan (Ogata,1987).



Gambar 2.3 Blok diagram pemodelan turbin angin (Sloth et al., 2011)

Daya yang dihasilkan oleh turbin angin (P) dipengaruhi oleh *mass rate* (\dot{m}), kecepatan angin (v), gaya gravitasi (g), massa jenis udara (ρ), dan luas area yang dilalui angin (A), yang dirumuskan sebagai berikut (Sami & Patton, 2012):

$$P = \dot{m} \frac{v_w^2}{2g} \quad (2.1)$$

$$\dot{m} = \rho A v_w \quad (2.2)$$

Apabila persamaan (2.2) disubstitusikan pada persamaan (2.1), maka diperoleh persamaan (2.3).

$$P = \frac{1}{2g} \rho A v_w^3 \quad (2.3)$$

Berdasarkan hukum mekanika rotasi, daya turbin angin dipengaruhi oleh nilai torsi generator (T) dan kecepatan rotor (ω). Secara sistematis, dituliskan seperti persamaan (2.4).

$$P = T\omega \quad (2.4)$$

Sedangkan daya yang dapat dikonversi oleh turbin angin dari energi angin, dapat dirumuskan pada persamaan (2.5).

$$P = \frac{1}{2} C_p \rho A v_w^3 \quad (2.5)$$

dengan nilai luasan (A) yang dilalui angin berupa lingkaran, maka nilai tersebut bergantung pada jari-jarinya (R).

$$A = \pi R^2 \quad (2.6)$$

Rotor pada turbin angin mengonversi energi kinetik dari angin menjadi energi untuk memutar bilah rotor dan poros, berputar pada kecepatan ω_r . Daya pada angin bergantung pada kecepatan angin, v_r , massa jenis udara, ρ , dan luas permukaan, A. Dari daya yang tersedia, daya yang didistribusikan menuju rotor bergantung pada nilai $C_p(\lambda, \beta)$, dimana β adalah fungsi sudut *pitch* pada baling-baling turbin, dan λ merupakan rasio antara kecepatan ujung baling-baling dan kecepatan angin. Persamaan torsi aerodinamik pada rotor, yaitu pada persamaan (2.7).

$$T_a = \frac{1}{2\omega_r} \rho A v_r^3 C_p(\lambda, \beta) \quad (2.7)$$

Selain rotor, salah satu komponen yang berperan penting pada sistem kerja turbin angin yaitu generator. Generator juga merupakan komponen turbin angin yang tidak terhubung secara langsung dengan turbin angin melainkan dengan *gearbox*. Penggunaan *gearbox* ini dikarenakan generator membutuhkan kecepatan putar yang jauh lebih besar dari kecepatan rotor turbin angin, sehingga dibutuhkan faktor pengali agar kecepatan putar generator bisa lebih tinggi. Pengaruh dari *gearbox* dengan faktor pengali n terhadap persamaan sistem dirumuskan pada persamaan (2.8) dan (2.9).

$$T_g = \frac{T_m}{n} \quad (2.8)$$

$$\omega_g = \omega_r n \quad (2.9)$$

Dengan,

- T_g : Torsi generator
- ω_g : Kecepatan generator
- T_m : Torsi mekanik generator
- n : Faktor pengali

Sehingga model matematis dari percepatan generator dapat dirumuskan pada persamaan (2.10).

$$2H \cdot \dot{\omega}_g = \theta_{\Delta} \cdot K_{sh} - (\omega_r - \omega_g) D_{mutual} - T_g - F \cdot \omega_g \quad (2.10)$$

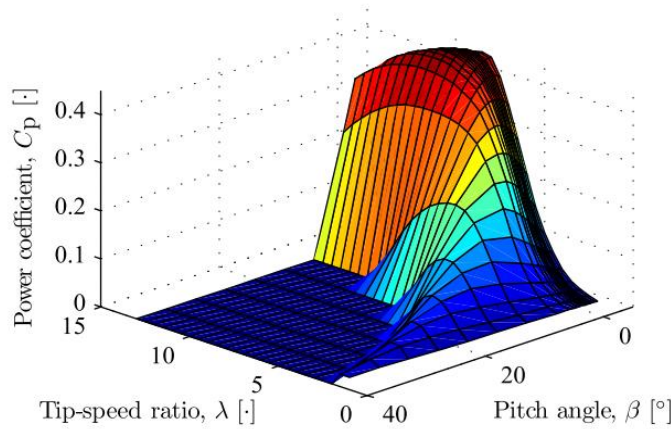
Dengan,

F : *Friction factor* generator

H : Konstanta inersia poros berkecepatan rendah

Setelah mendapatkan semua model matematis turbin angin sebenarnya ada satu variabel yang sangat penting yaitu koefisien daya yang mempengaruhi konversi daya dari turbin angin dikarenakan koefisien daya memiliki nilai maksimal yaitu 59% dikarenakan tidak ada satupun turbin angin yang dapat mengkonversi energi kinetik menjadi energy mekanik rotor. Didalam persamaan (2.5) terdapat satu variabel yang memiliki peran penting dalam perhitungan daya maupun torsi yaitu C_p dimana C_p memiliki nilai bervariasi salah satunya terhadap *tip speed ratio* yang akan ditunjukkan pada Gambar 2.4 dan persamaannya ditunjukkan pada persamaan (2. 11).

$$C_p(\lambda, \beta) = C_1 \left(\frac{C_2}{\lambda_i} - C_3 \beta - C_4 \right) e^{\frac{-C}{\lambda_i}} + C_6 \lambda \quad (2.11)$$



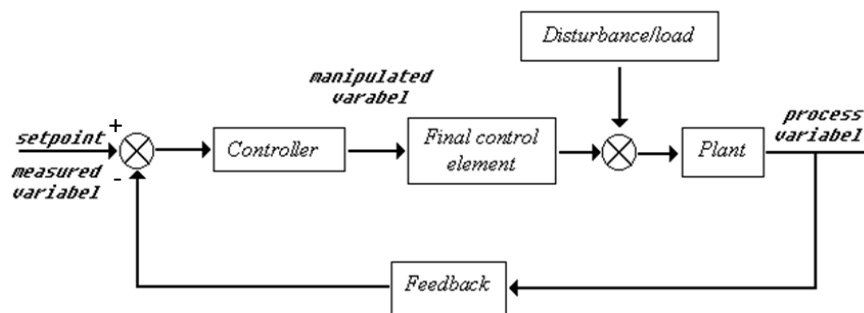
Gambar 2.4 Ilustrasi koefisien daya, C_p (Sloth et al., 2011)

2.4. Sistem Kendali PI

Sistem pengendalian adalah gabungan dari kerja suatu komponen yang digunakan untuk mempertahankan variabel yang dikendalikan (*process variable*) pada suatu nilai tertentu (*set point*) sehingga sistem dapat dikatakan stabil. Seiring perkembangan ilmu pengetahuan dan teknologi suatu sistem pengendalian sudah memakai unit kendali otomatis seperti pada Gambar 2.5. Algoritma pengendali *Proportional-Integral-Derivative* (PID) hingga saat ini

dapat dikatakan sebagai algoritma pengendali terbaik karena keefektifannya, sederhana dalam implementasi dan luas penggunaannya. Karakteristik pengendali PID sangat dipengaruhi oleh kontribusi besaran dari ketiga parameter P, I dan D. Penyetelan konstanta K_p , T_i dan T_d akan mengakibatkan menonjol sifat dari masing – masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat diatur lebih menonjol dibandingkan dengan yang lain, konstanta yang menonjol itulah yang akan memberikan pengaruh pada respon sistem secara keseluruhan.

Pengendali PI merupakan penggabungan antara dua macam pengendali, yaitu P (*Propotional*) dan I (*Integral*). Pengendali PI merupakan jenis pengendali untuk menentukan suatu kestabilan atau kepresisian sistem instrumentasi dengan adanya *feedback* atau umpan balik pada sistem tersebut. Perancangan sistem kendali PI bertujuan untuk mengetahui respon dari sistem jika ditambahkan dengan pengendali yang telah didapatkan nilai *gain*nya serta untuk mengetahui kesesuaian kerja sistem secara keseluruhan.



Gambar 2.5 Diagram blok sistem pengendalian tertutup (Basilio, 2015)

Persamaan untuk pengendali PI adalah sebagai berikut :

$$u(t) = K_p \cdot e(t) + \frac{1}{K_i} \int e dt \quad (2.12)$$

dengan,

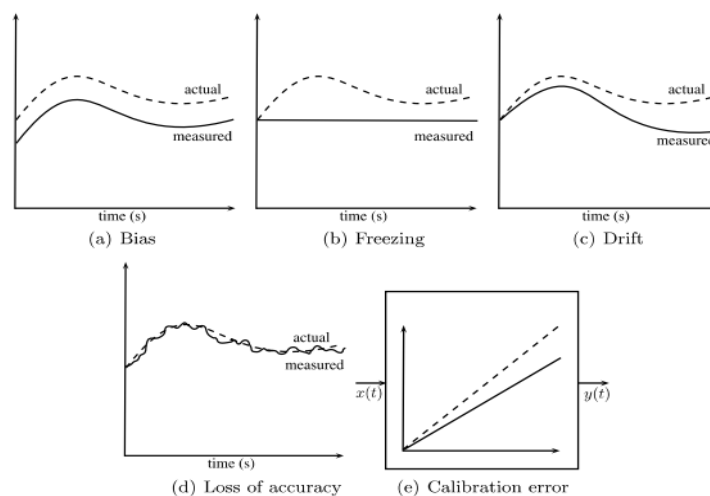
$u(t)$: input pengendali
 K_p : konstanta proposional
 $e(t)$: sinyal error
 K_i : konstanta integral

2.5. Kesalahan Sistem Kontrol

Istilah kesalahan dan kegagalan digunakan untuk menjelaskan tingkat degradasi sistem. Menurut (Isermann, 2006), kesalahan didefinisikan sebagai

penyimpangan yang tak diinginkan atas paling tidak satu sifat karakteristik atau parameter sistem dari kondisi standar/biasa/yang dapat diterima. Kegagalan didefinisikan sebagai interupsi permanen atas kemampuan sistem untuk melakukan fungsi yang diperlukan pada kondisi operasi tertentu. Sehingga kegagalan lebih berat kerusakannya dari pada kesalahan.

Kesalahan sensor dapat disebabkan kalibrasi yang buruk atau bias dan perubahan karakteristik dinamik yang menyebabkan *error* pada keluaran sensor namun bukan pada dinamika plant. Beberapa literatur juga mengklasifikasikan kesalahan berdasarkan pemodelan sinyalnya, yaitu kesalahan *additive* dan *multiplicative* (Isermann, 2006). Kesalahan *additive* mempengaruhi sinyal dengan sebuah penambahan sinyal kesalahan, sementara kesalahan *multiplicative* dengan sebuah perkalian sinyal kesalahan. Kesalahan *additive* memberikan deskripsi terbaik atas sebuah kesalahan pada kasus dimana kesalahan tidak bergantung pada masukan dan/atau *state* sistem. Sedangkan kesalahan *multiplicative*. Kesalahan *multiplicative* tidak ditinjau dalam penelitian thesis ini.



Gambar 2.6 Jenis kesalahan pada sensor (Bruno et al., 2003)

Dalam (H, Alwi et al., 2016) disebutkan beberapa jenis kesalahan sensor yang biasa ditemukan, yaitu: *bias* (*error* konstan pada output), *freezing* (sensor tidak merespon input), *drift* (penambahan pada deviasi output), *loss of accuracy* (berkurangnya akurasi), dan *calibration error* (berkurangnya fungsi fisik pembacaan sensor).

2.6. Karakteristik Sistem Pengukuran

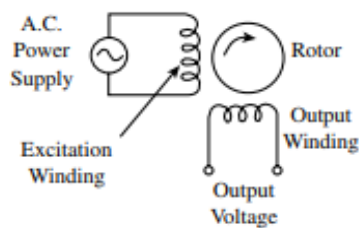
Akurasi dan performansi sistem pengukuran sangat ditentukan oleh karakteristik instrumen yang digunakan didalamnya. Pengetahuan tentang karakteristik instrumen sangat penting ketika merancang sistem pengukuran untuk memastikan bahwa persyaratan pengukuran terpenuhi dan instrumen yang paling tepat digunakan. Karakteristik statik merupakan parameter instrumen dalam keadaan tunak, ketika output telah ditetapkan ke pembacaan *steady*. Karakteristik ini memiliki dampak mendasar pada kualitas pengukuran yang diperoleh.

Sensor sering didefinisikan sebagai perangkat yang menerima stimulus dan merespon dengan sinyal listrik. Stimulus berupa suatu keadaan tertentu pada suatu lingkungan yang akan dideteksi maupun ditangkap oleh sensor (Fraden, 2016). Pengertian sensor dalam sistem pengukuran merupakan elemen pertama yang memberikan sinyal keluaran sebagai fungsi dari masukan besaran fisis tertentu (Bentley, 2005). Karakteristik sensor merupakan performansi dari sebuah sensor. Salah satu karakteristik sensor adalah karakteristik statik yang meliputi *span*, *range*, linieritas, sensitivitas, resolusi dan histeresis (Fraden, 2016).

Span merupakan jangkauan pengukuran dimana sensor masih dapat merespon stimulus yang diberikan. Jangkauan pengukuran sensor terletak pada masukan maupun keluaran sensor. Jangkauan pengukuran masukan dihitung dari selisih masukan maksimum dan masukan minimum, sedangkan jangkauan pengukuran keluaran juga dihitung dari selisih keluaran maksimum dan keluaran minimum. *Range* pengukuran masukan merupakan nilai masukan minimum sampai masukan minimum. *Range* keluaran adalah nilai minimum keluaran sampai keluaran maksimum. Linieritas merupakan karakteristik ideal dari sistem pengukuran. Sensitivitas akan menunjukkan seberapa jauh kepekaan suatu sensor. Kepekaan suatu sensor dapat diketahui dari perbandingan perubahan keluaran terhadap perubahan masukan. Resolusi merupakan adanya perubahan nilai masukan terbesar sensor tidak akan menyebabkan perubahan pada nilai keluaran sensor. Histeresis adalah selisih nilai keluaran sensor yang dapat diperoleh dari pengukuran naik dan pengukuran turun terhadap nilai masukan yang sama.

2.7. Tachometer

Tachometer merupakan sebuah perangkat untuk mengukur kecepatan rotasi poros mesin (Stephen A, 2001). Jenis tachometer yang banyak digunakan adalah tachometer DC. Tachometer DC adalah sebuah generator DC yang memproduksi tegangan keluaran DC yang proporsional dengan kecepatan batang. Terdiri dari magnet permanen dan bagian yang berputar terbuat dari koil, dan juga terjadi konversi langsung. Prinsip kerjanya proses konversi langsung antara kecepatan dan tegangan. Tegangan yang diinduksi dalam lilitan adalah sinusoidal, komutator dan sikat pada rotor mengubah tegangan AC pada lilitan ke output DC dengan cara yang sama seperti di DC generator. Kesalahan sensitivitas pada tachometer biasanya disebabkan karena keausan sikat pada rotor yang digunakan untuk menjaga inersia (Brodgesell, 2003). Tachometer DC ditampilkan secara skematis pada Gambar 2.7.



Gambar 2.7 Skema Tachometer DC (Brodgesell, 2003)

2.8. Observer

Dalam pendekatan *pole-placement* pada perancangan sistem kontrol, kita mengasumsikan bahwa semua variabel keadaan tersedia untuk umpan balik. Namun, dalam praktiknya, tidak semua variabel keadaan tersedia untuk umpan balik. Maka kita perlu memperkirakan variabel keadaan yang tidak tersedia. Estimasi variabel keadaan yang tidak terukur biasanya disebut observasi. Perangkat (atau sebuah program komputer) yang memperkirakan atau mengamati variabel keadaan disebut *state observer*, atau bisa disebut seorang *observer*. Jika *state observer* mengamati semua variabel keadaan sistem, terlepas dari apakah beberapa variabel keadaan tersedia untuk pengukuran langsung, ini disebut *full order state observer*. Ada kalanya hal ini tidak diperlukan, kapan kita memerlukan observasi hanya variabel

keadaan yang tidak terukur, tapi bukan dari yang sebenarnya bisa diukur secara langsung juga. Sebagai contoh, karena variabel *output* dapat diamati dan berhubungan secara linear dengan variabel keadaan, kita tidak perlu mengamati semua variabel keadaan, namun hanya mengamati variabel *n-state*, di mana *n* adalah dimensi vektor keadaan dan *m* adalah dimensi dari vektor keluaran (Ogata, 2004).

Observer yang memperkirakan kurang dari *n* variabel keadaan, dimana *n* adalah dimensi dari vektor keadaan, disebut *reduced-order state observer* atau secara sederhana disebut *reduced-order observer*. Jika observer dengan tingkat reduksi minimal, maka observer disebut *minimum-order state observer* atau *minimum-order observer* (Ogata, 2004).

Berikut adalah model yang ditunjukkan oleh persamaan (2.13) dan (2.14)

$$x(k+1) = Ax(k) + Bu(k) \quad (2.13)$$

$$y(k) = Cx(k) + Du(k) \quad (2.14)$$

dengan:

A : Matriks keadaan (*state*)

B : Matriks kendali

C : Matriks keluaran

D : Matriks gangguan

$u(k)$: Masukan kendali

$x(k+1)$: Turunan vektor keadaan

$y(k)$: Vektor keluaran

$x(k)$: Vektor keadaan (*state*)

Observer adalah *subsystem* untuk merekonstruksi vektor *state* dari plant. Model matematis dari *observer* pada dasarnya sama dengan plant kecuali kita memasukkan syarat tambahan yang didalamnya terdapat perkiraan *error* untuk mengkompensasi ketidakakuratan dalam matriks A dan B serta *error* awal. Perkiraan *error* atau *error* observasi adalah perbedaan diantara output hasil pengukuran dengan estimasi output. *Error* awal adalah perbedaan diantara *state* awal dengan estimasi *state*. Sehingga model matematis dari *observer* menjadi persamaan (2.15) dan (2.16) (Ogata, 2004).

$$\hat{x}(k+1) = A.\hat{x}(k) + B.u(k) + K_e(y(k) - \hat{y}(k)) \quad (2.15)$$

$$\hat{y}(k) = C\hat{x}(k) \quad (2.16)$$

dengan,

$\hat{x}(k+1)$: Estimasi *state* $x(k+1)$

K_e : *Gain Observer*

$\hat{x}(k)$: Estimasi *state* $x(k)$

$\hat{y}(k)$: Estimasi $y(k)$

$y(k)$ pada persamaan (2.13) dan $\hat{y}(k)$ pada (2.16) disubstitusikan ke dalam persamaan (2.14), maka diperoleh

$$\hat{x}(k+1) = A.\hat{x}(k) + B.u(k) + K_e(Cx(k) - C\hat{x}(k)) \quad (2.17)$$

Persamaan *error observer* atau $(e(k+1))$ didapatkan dari hasil pengurangan persamaan (2.17) dengan persamaan (2.12), menjadi :

$$e(k+1) = x(k+1) - \hat{x}(k+1) \quad (2.18)$$

$$e(k+1) = Ax(k) - A\hat{x}(k) - K_e(Cx(k) - C\hat{x}(k)) \quad (2.19)$$

$$e(k+1) = (A - K_eC)(x(k) - \hat{x}(k)) \quad (2.20)$$

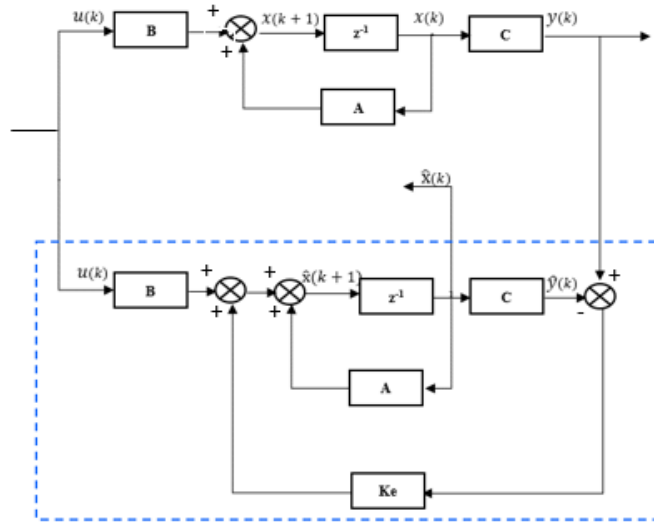
$$e(k+1) = (A - K_eC)e(k) \quad (2.21)$$

Berdasarkan Gambar 2.8, *observer* mengandung nilai *gain observer* atau K_e . *Gain observer* merupakan matriks untuk mengoreksi perbedaan antara keluaran y aktual dan y estimasi sehingga dapat digunakan untuk mengestimasi *state* yang tidak terukur (Ogata, 2004). Untuk menentukan nilai K_e sangat dipengaruhi oleh nilai letak *pole*, seperti pada Gambar 2.19 untuk sistem dengan domain diskrit, posisi nilai *pole* untuk menyatakan sistem stabil berada tidak lebih dari 1 atau -1 pada sumbu imajiner maupun sumbu riil. Nilai K_e diperoleh berdasarkan persamaan (2.21), dengan

$$e(k) = (x(k) - \hat{x}(k)) \quad (2.22)$$

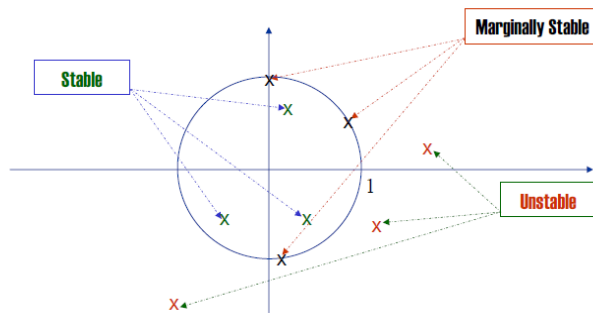
Kemudian dari persamaan (2.21) menjadi persamaan karakteristik observer yang dituliskan seperti persamaan (2.23).

$$|zI - A + K_e C| = 0 \quad (2.23)$$



Gambar 2.8 *Observer* (Ogata, 2004)

Maka dengan menentukan nilai p atau posisi *pole*, diperoleh persamaan karakteristik yang diinginkan, maka nilai K_e diperoleh dari mensubstitusikan persamaan (2.23) ke persamaan karakteristik yang diinginkan (Ogata, 2004).



Gambar 2.9 Kestabilan pole *placement* sistem diskrit (M. Chen, 2016)

2.9. Algoritma *Exponential Smoothing* untuk Prediksi Kesalahan

Kesalahan yang dimodelkan sebagai *output* aditif dari sistem turbin angin merupakan kesalahan sensor. Dengan merancang *observer* maka dapat menghasilkan estimasi kesalahan sensor. Estimasi kesalahan sensor diasumsikan linier dari indeks waktu diskrit n . Metode *Holt's* diadopsi untuk memproses

prediksi kesalahan. Algoritma prediksi kesalahan dapat dihitung melalui *exponential smoothing algorithm* sebagai berikut (Xu et al., 2009):

$$u_n = a\hat{a}_n + (1 - a)(u_{n-1} + s_{n-1}) \quad (2.24)$$

$$s_n = b(u_n - u_{n-1}) + (1 - b)s_{n-1} \quad (2.25)$$

Misalkan u_n adalah hasil estimasi yang diperhalus dan s_n adalah faktor pertumbuhan linear yang diestimasi. Diberikan hasil estimasi kesalahan $\hat{a}_n (n = 1, 2, \dots)$ yang diperoleh dari *observer*. Dimana $0 < a < 1$ & $0 < b < 1$ adalah parameter *Holt's method* (Xu et al., 2009). Kondisi awal untuk *Holt's method* sebagai berikut:

$$u_1 = \hat{a}_1 = \hat{f}(k) \quad (2.26)$$

$$u_2 = \hat{a}_2 = \hat{f}(k + 1) \quad (2.27)$$

$$s_2 = u_2 - u_1 \quad (2.28)$$

Setelah menghitung estimasi kesalahan pada saat nT dan estimasi faktor pertumbuhan, maka prediksi kesalahan dapat dihitung menggunakan algoritma berikut (Xu et al., 2009):

$$\bar{a}_{n+l} = u_n + s_n lT \quad (2.29)$$

Dengan

l : step prediksi

T : *time interval* untuk prediksi kesalahan

Penjelasan mengenai algoritma prediksi kesalahan adalah sebagai berikut :

- 1) Step estimasi kesalahan yang diperhalus: Jika $n = 1$, atau $n = 2$, menghitung estimasi kesalahan yang diperhalus u_n menggunakan persamaan (2.26), atau (2.27). Jika $n \geq 3$ menghitung estimasi kesalahan yang diperhalus u_n menggunakan persamaan (2.24).
- 2) Step estimasi faktor pertumbuhan: Menghitung n -step prediksi faktor pertumbuhan s_n dengan menggunakan (2.28) jika $n = 2$, dan menghitung n -step estimasi faktor pertumbuhan s_n dengan menggunakan (2.25) jika $n \geq 3$.
- 3) Step prediksi kesalahan: Menghitung prediksi kesalahan \bar{a}_{n+l} dengan menggunakan persamaan (2.29).

2.10. Keandalan

Keandalan didefinisikan sebagai kemampuan suatu sistem atau komponen untuk melakukan fungsi yang diperlukan dalam kondisi yang ditentukan untuk waktu yang ditentukan (P. O'Connor, 2012). Menurut Dhillon (2005) keandalan adalah kemungkinan untuk tidak mengalami kegagalan atau dapat melaksanakan fungsinya selama periode waktu (t). Evaluasi keandalan dapat membantu dalam memperkirakan peluang sebuah sistem atau komponen untuk dapat melaksanakan fungsinya dalam jangka waktu tertentu dan dalam kondisi tertentu. Sistem atau komponen masih dapat beroperasi walaupun tidak dapat memenuhi fungsinya secara maksimal.

Nilai fungsi *reliability* merupakan nilai probabilitas, maka nilai fungsi *reliability* (R) bernilai $0 \leq R \leq 1$ (Ebeling, 1997). Keandalan turbin angin bergantung pada keandalan beberapa kunci komponen. Misalnya, kegagalan *gearbox* terlihat sebagai kegagalan paling umum dan paling kritis. Kegagalan *gearbox* menghentikan fungsi utama dari pembangkit listrik dan menghadapi *downtime* untuk perbaikan atau penggantian. Kerusakan struktural pada baling-baling dan menara merupakan kegagalan pada umumnya. Mata pisau mungkin mengalami beban angin yang tinggi atau burung menabrak dengan keras sehingga baling-baling putus (Chan & Mo, 2017). (Chou & Tu, 2011) menganalisis kegagalan menara turbin angin yang disebabkan oleh *Typhoon Jangmi*. Mereka menyimpulkan bahwa baut tidak memadai dan kontrol kualitas adalah penyebab kerusakan.

2.11. Prediksi Keandalan *Real-Time*

Prediksi akan selalu bersangkutan ketika melakukan evaluasi pada sebuah sistem. Hal ini dilakukan sebagai pertimbangan sebelum melakukan sebuah keputusan untuk membangun atau menggunakan sebuah sistem. Prediksi akan lebih berkaitan dengan analisa berdasarkan model dibandingkan dengan menggunakan data sebenarnya dan menyediakan dasar untuk pengujian, pembangunan dan evaluasi dari pada keandalan, kemudian dari pada itu terdapat juga bagian perawatan dan pengaturan lainnya (Blischke W & Prabhakar M, 2000).

Penelitian mengenai keandalan *real-time* atau yang biasanya disebut juga dengan keandalan *on-line* ini mulai berkembang sejak tahun 1990-an. Pada tahun 1992, (Taylor et al., 2007) meneliti mengenai prediksi kendalan *real-time* kondisional berdasarkan data kinerja alat. Sedangkan pada 1999, (Chinnam, 1999) melakukan penelitian mengenai estimasi keandalan *on-line* dari masing-masing komponen menggunakan sinyal degradasi. Pada 2002, (Chinnam & Mohan, 2002) kembali melanjutkan penelitiannya dengan mengambil topik berupa estimasi keandalan *on-line* dari sistem fisik menggunakan jaringan saraf. Penelitian tersebut terus berkembang hingga pada tahun 2009, (Xu et al., 2009) memperkenalkan penelitiannya mengenai sebuah metode baru dalam memprediksi keandalan *real-time* untuk sistem dinamik yang digabungkan dengan algoritma prediksi kesalahan.

Setelah kesalahan diprediksi, keandalan pada sistem dinamik untuk beberapa waktu ke depan dapat diprediksi. Untuk memprediksi keandalan sistem, ada beberapa hal penting yang harus diperhatikan. Pertama, selain dari definisi kegagalan sebagai nilai kesalahan yang telah mencapai batas, kegagalan sistem juga didefinisikan sebagai keadaan sistem yang telah memasuki area terlarang \mathcal{M}_1 . Kedua, keadaan sistem tidak dapat diukur secara langsung, harus melalui estimasi dikarenakan kegagalan sistem didefinisikan dalam sebuah keadaan. Keadaan diestimasi menggunakan algoritma OTK-SIR (Xu et al., 2009). Pada penelitian ini, penulis tidak menggunakan algoritma OTK-SIR, namun menggunakan algoritma *observer*. Ketiga, kunci dari prediksi keandalan adalah prediksi kesalahan (2.29).

$$R(nT, (n+r)T) = \Pr(x_{n+r} \in \mathcal{M}_0 | x_n \in \mathcal{M}_0) \quad (2.30)$$

Penjelasan mengenai algoritma RP (*Reability Prediction*) adalah sebagai berikut :

- 1) Step inisialisasi : l yang telah dinotasikan pada algoritma prediksi kesalahan dijadikan $l = 1$. Inisialisasi partikel menjadi $\bar{X}_{n+l-1|n}^{(i)} = \bar{X}_{n+0|n}^{(i)} = \hat{X}_{n|n}^{(i)}$, $i = 1, \dots, N$.
- 2) Step estimasi kesalahan : Jalankan algoritma *observer* untuk menghasilkan estimasi kesalahan \hat{a}_1 dan \hat{a}_2 pada persamaan (2.26) dan (2.27).

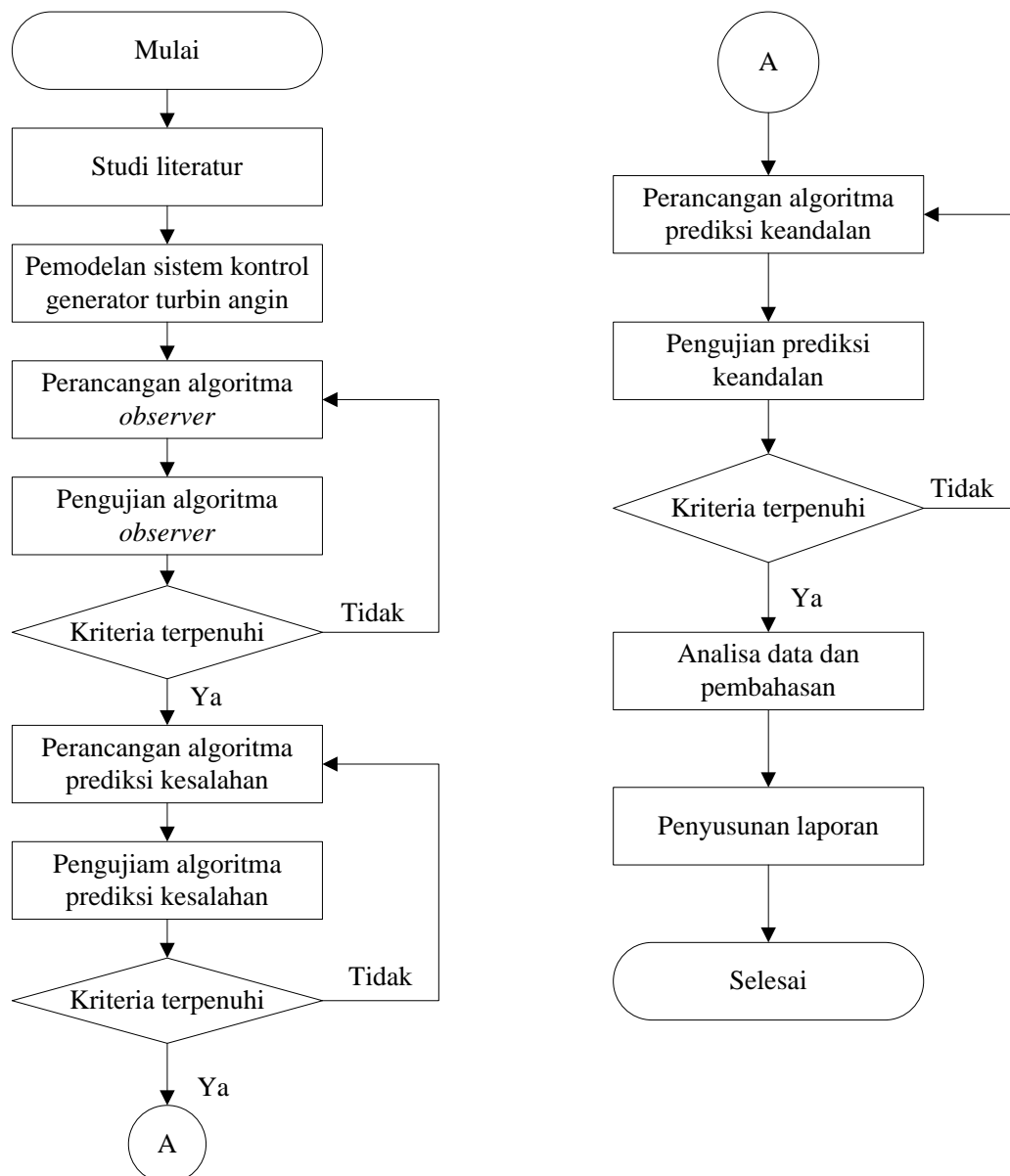
- 3) Step prediksi kesalahan: Jalankan algoritma prediksi kesalahan dengan langkah untuk menghitung prediksi kesalahan $\tilde{\alpha}_{n+l}$.
- 4) Step prediksi partikel : untuk $i = 1, \dots, N$, hitung partikel baru $\bar{X}_{n+l-1|n}^{(i)}$ dari sistem persamaan $\bar{X}_{n+l-1|n}^{(i)} = f(\bar{X}_{n+l-1|n}^{(i)}) + \Gamma \tilde{\alpha}_{n+l} + W_{n+l}^{(i)}$ dimana $W_{n+l}^{(i)}$ adalah realisasi dari w_{n+l} untuk partikel ke- i .
- 5) Step pemilihan partikel : jika $l < r$, maka $l = l + 1$, dan kembali ke step 2, jika sebaliknya lanjut pada step 6.
- 6) Step perhitungan prediksi keandalan : hitung jumlah partikel dari himpunan $\{\bar{X}_{n+r|n}^{(i)}\} (i = 1, \dots, N)$ yang jatuh pada \mathcal{M}_0 dan dinotasikan oleh N_0 . Maka sistem prediksi keandalan selama $(nT, (n+r)T]$ dihitung oleh N_0/N .

Halaman ini sengaja dikosongkan

BAB 3

METODA PENELITIAN

Prosedur penelitian terdiri dari beberapa tahapan yang dilakukan dari awal hingga akhir untuk tercapainya tujuan dari penelitian ini. Prosedur penelitian secara garis besar digambarkan dalam diagram alir di bawah ini :



Gambar 3.1 Diagram Alir Penelitian

Prosedur penelitian dibagi menjadi beberapa tahapan yang akan dijelaskan sebagai berikut :

3.1 Studi Literatur

Studi literatur merupakan tahapan awal untuk menunjang terselesainya penelitian ini. Literatur yang dapat digunakan antara lain buku teks/ *handbook*, prosiding seminar, jurnal penelitian, artikel ilmiah dan laporan tesis maupun disertasi dibutuhkan untuk mengetahui perkembangan terkini yang berkaitan dengan penelitian sejenis yang telah dilakukan para peneliti sebelumnya. Kegiatan yang dilakukan dalam studi literatur yaitu pengayaan materi yang berkaitan dengan topik prediksi keandalan. Studi literatur mengenai prediksi keandalan yang digunakan untuk sistem generator turbin angin. Teori yang dipelajari antara lain algoritma *observer*, algoritma prediksi kesalahan, algoritma prediksi keandalan dan mekanisme perancangan sistem prediksi keandalan pada sistem pengendalian kecepatan turbin angin secara real-time berbasis prediksi kesalahan sensor.

3.2 Pemodelan Sistem Kontrol Generator Turbin Angin

Pemodelan sistem turbin angin merupakan langkah kedua dalam penelitian ini. Spesifikasi turbin angin akan ditunjukkan pada Tabel 3.1.

Tabel 3.1 Spesifikasi Turbin Angin Pada Simulasi (J. F. Manwell & J. G. McGowan, 2009)

<i>Model</i>	<i>Horizontal Axis Wind Turbine (HAWT)</i>
<i>Diameter rotor (m)</i>	2.56
<i>Swept area(m²)</i>	5.15
<i>Rated wind speed (m/s)</i>	11
<i>Output Power (watt)</i>	400
<i>Maximum design wind speed (pu/rad)</i>	80.27
<i>Blade Material</i>	Polypropylene
<i>Battery System Voltages (Volt)</i>	12, 24 or 48

Pada penelitian ini, model sistem turbin angin yang digunakan adalah dalam satuan per unit (pu). Nilai satuan pu adalah nilai yang terkait dengan jumlah dasar. Dalam analisis sistem tenaga, nilai satuan pu umum digunakan untuk menganalisis dan mengkomunikasikan nilai voltase, arus, daya dan impedansi. Jumlah satuan pu ini dinormalisasi atau diskalakan pada basis yang dipilih, seperti yang ditunjukkan pada persamaan (3.1) yang memungkinkan para insinyur menyederhanakan perhitungan sistem tenaga dengan beberapa transformasi tegangan. Parameter-parameter yang digunakan dalam pemodelan turbin angin ini mengacu pada (Nicholas *et al.*, 2003). Nilai-nilai parameter tersebut dapat kita lihat pada Tabel 3.2 berikut.

Tabel 3.2 Parameter – parameter pada pemodelan turbin angin (Nicholas *et al.*, 2003)

Parameter	Nilai	Satuan
Sudut blade pitch maksimum (β_{max})	27	deg
Sudut blade pitch minimum (β_{min})	0	deg
Pitch rate $\beta/detik(deg/detik)$	10	deg/detik
Tip speed ratio maksimum (λ_{max})	9.95	-
Coefficient of power maksimum (Cp_{max})	0.5	-
Torsi mula-mula (T_0)	0.83	pu
Koefisien kekakuan (K_{sh})	80.27	pu/rad
Friction factor of generator (F)	0.01	pu
Kecepatan putar maksimum (ω_{max})	1.2	pu
Kecepatan putar minimum (ω_{min})	0.5	pu
Konstanta inersia poros berkecepatan tinggi (H_{wt})	4.32	detik
Konstanta inersia poros berkecepatan rendah (H)	0.685	detik
Konstanta Redaman (D_{mutual})	1.5	pu

Setelah mendapatkan bentuk plant dari turbin angin maka dilakukan pemodelan turbin angin agar dapat disimulasikan. Untuk mendefinisikan nilai C_p yang bervariasi dari C_1 sampai dengan C_8 , maka digunakan nilai yang telah didapatkan pada (Nicholas,dkk. 2003) sebagai berikut :

$$C_1 = 0.645$$

$$C_2 = 116$$

$$C_3 = 0.4$$

$$C_4 = 5$$

$$C_5 = 21$$

$$C_6 = 0.00912$$

$$C_7 = 0.08$$

$$C_8 = 0.035$$

Pemodelan *drive train* pada turbin angin mencakup percepatan rotor $\dot{\omega}_r$ dan sudut torsi $\dot{\theta}_\Delta$. Persamaan turunan sudut torsi $\dot{\theta}_\Delta$ ditunjukkan pada persamaan (3.2).

$$\text{Nilai per unit} = \frac{\text{Nilai sebenarnya (actual quality)}}{\text{Nilai dasar nominal (base quality)}} \quad (3.1)$$

$$\dot{\theta}_\Delta = \omega_r - \omega_g \quad (3.2)$$

Turbin angin dikarakteristikan berdasarkan perilaku nonlinearnya dan ketergantungannya pada kekuatan angin stokastik yang tak terkendali sebagai sinyal penggerak. Pada umumnya, pemodelan turbin angin dilakukan dengan menggabungkan model subsistem turbin angin sehingga bersama-sama membentuk dinamika pemodelan turbin angin secara keseluruhan. Pemodelan turbin angin dalam satuan pu dapat dilakukan dengan menggunakan persamaan persamaan berikut :

- Pemodelan Penggerak (*Drive Train*)

Penggerak pada turbin angin terdiri dari poros berkecepatan tinggi dan poros berkecepatan rendah, dimana memiliki inersia H_{wt} dan H . Selain itu juga terdapat konstanta pegas, K_{sh} , dan konstanta redaman, D_{mutual} . Poros-poros yang saling berhubungan melalui transmisi memiliki rasio gigi n_g , dikombinasikan dengan nilai sudut torsi, θ_Δ , dan torsi generator,

T_g , pada kecepatan ω_g . Persamaan rotor dalam satuan pu dapat dituliskan seperti pada persamaan (3.3).

$$\dot{\omega}_r = \frac{T_a - T_m}{2H_{wt}} \quad (3.3)$$

dengan,

H_{wt} : Konstanta inersia poros berkecepatan tinggi

Torsi mekanik rotor, T_m selanjutnya dimodelkan pada persamaan (3.4).

$$T_m = \theta_{\Delta} \cdot K_{sh} + (\omega_r - \omega_g) D_{mutual} \quad (3.4)$$

dengan,

K_{sh} : Koefisien kekakuan

D_{mutual} : Koefisien redaman

Persamaan (3.3) disubstitusikan kedalam persamaan (3.4), sehingga didapatkan persamaan (3.5) berikut.

$$\dot{\omega}_r = \frac{T_a - (\omega_r - \omega_g) D_{mutual} - \theta_{\Delta} \cdot K_{sh}}{2H_{wt}} \quad (3.5)$$

Sedangkan persamaan generator dalam satuan pu dapat dituliskan seperti pada persamaan (3.6).

$$\dot{\omega}_g = \frac{\theta_{\Delta} \cdot K_{sh} - (\omega_r - \omega_g) D_{mutual} - T_g - F \cdot \omega_g}{2H} \quad (3.6)$$

dengan,

F : *Friction factor* generator

H : Konstanta inersia poros berkecepatan rendah

Persamaan sudut torsi dalam satuan pu dapat dituliskan seperti pada persamaan (3.7).

$$\dot{\theta}_{\Delta} = \omega_r - \frac{\omega_g}{n_g} \quad (3.7)$$

dengan,

n_g : Rasio gigi

- Pemodelan Generator dan Konverter

Tahap berikutnya yaitu memodelkan *actuator* yang akan digunakan. Aktuator berupa *converter*. Generator dimodelkan hanya berupa model mekanik, maka diasumsikan konverter memiliki *controller* internal sehingga converter membebani generator dengan torsi tertentu. Sehingga model yang disederhanakan dari *converter* dapat didekati dengan sistem orde satu. Torsi generator pada persamaan (3.8) dikontrol oleh torsi referensi, $T_{g,ref}$. Dinamika konverter diestimasi oleh sistem melalui sistem orde pertama dengan waktu konstan τ_g . Sehingga didapatkan persamaan berikut :

$$\dot{T}_g = -\frac{1}{\tau_g}T_g + \frac{1}{\tau_g}T_{g,ref} \quad (3.8)$$

dengan,

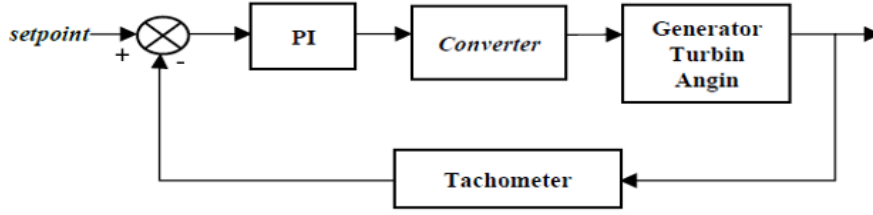
T_g : Torsi generator

$T_{g,ref}$: Torsi referensi

τ_g : Waktu konstan

Pengendali PI pada simulasi digunakan untuk mengatasi karakteristik nonlinieritas yang diakibatkan oleh sistem turbin angin. Pengendalian ini menggunakan parameter proporsional (P) yang bertindak sebagai gain dan parameter integral (I) yang bertindak sebagai integrator untuk menghilangkan *offset* pada respon sistem.

Karakteristik pengendali PI sangat dipengaruhi oleh kontribusi besaran dari ketiga parameter P, dan I. Penyetelan konstanta K_p dan T_i akan mengakibatkan penonjolan sifat dari masing – masing elemen. Diagram blok sistem pengendalian kecepatan turbin angin ditunjukkan oleh Gambar 3.2. Dalam pengendalian kecepatan turbin angin dilakukan perancangan sistem kendali PI. Nilai K_p , K_i didapatkan secara eksperimen dengan metode *Ziegler–Nichols* untuk perancangan secara simulasi dengan *simulink* pada perangkat lunak MATLAB R2104a. Simulasi dilakukan sampai sistem kendali dapat bekerja dengan baik.



Gambar 3.2 Diagram Blok Sistem Pengendalian Kecepatan Turbin Angin

Perancangan sistem kendali PI bertujuan untuk mengetahui respon dari sistem jika ditambahkan dengan pengendali yang telah didapatkan nilai *gain*nya serta untuk mengetahui kesesuaian kerja sistem secara keseluruhan. Sistem yang dirancang telah bekerja dengan baik jika respon sistem dapat mencapai nilai *setpoint* yang telah diberikan. Perancangan pengendali PI dengan menggunakan *simulink* pada perangkat lunak matlab R2014a dapat dilihat pada lampiran.

3.3 Perancangan Algoritma *Observer*

Perancangan *observer* digunakan untuk menghasilkan estimasi kesalahan pada sistem generator turbin angin. Untuk itu diperlukan model ruang keadaan yang dapat mempresentasikan *plant* generator turbin angin dengan benar. Selanjutnya parameter *observer* diturunkan dari model ruang keadaan ini. *Gain observer* diperoleh dengan teknik *extended state observer* (ESO) seperti yang dilakukan dalam (Indriawati et al., 2015), tetapi perubahan dilakukan dari bentuk kontinyu ke dalam bentuk domain diskrit. Berdasarkan persamaan (3.8) maka disusunlah persamaan matriks turbin angin dalam bentuk berikut:

$$\begin{bmatrix} \dot{T}_g \\ \dot{\omega}_r \\ \dot{\omega}_g \\ \dot{\theta}_\Delta \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_g} & 0 & 0 & 0 \\ 0 & -\frac{D_m}{2H_{wt}} & \frac{D_m}{2H_{wt}} & -\frac{K_{sh}}{2H_{wt}} \\ -\frac{1}{2H} & \frac{D_m}{n_g \cdot 2H} & -\frac{(D_m+F)}{n_g^2 \cdot 2H} & \frac{K_{sh}}{n_g^2 \cdot 2H} \\ 0 & 1 & -\frac{1}{n_g} & 0 \end{bmatrix} \begin{bmatrix} T_g \\ \omega_r \\ \omega_g \\ \theta_\Delta \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{\tau_g} \\ \frac{1}{2H_{wt}} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_a \\ T_{g,ref} \end{bmatrix} \quad (3.9)$$

dengan,

$$x(k+1) = \begin{bmatrix} \dot{T}_g \\ \dot{\omega}_r \\ \dot{\omega}_g \\ \dot{\theta}_\Delta \end{bmatrix}; A = \begin{bmatrix} -\frac{1}{\tau_g} & 0 & 0 & 0 \\ 0 & -\frac{D_m}{2H_{wt}} & \frac{D_m}{2H_{wt}} & -\frac{K_{sh}}{2H_{wt}} \\ -\frac{1}{2H} & \frac{D_m}{n_g \cdot 2H} & -\frac{(D_{ref} + F)}{n_g^2 \cdot 2H} & \frac{K_{sh}}{n_g^2 \cdot 2H} \\ 0 & 1 & -\frac{1}{n_g} & 0 \end{bmatrix}$$

$$x(k) = \begin{bmatrix} T_g \\ \omega_r \\ \omega_g \\ \theta_\Delta \end{bmatrix}; B = \begin{bmatrix} 0 & \frac{1}{\tau_g} \\ \frac{1}{2H_{wt}} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; u(k) = \begin{bmatrix} T_a \\ T_{g,ref} \end{bmatrix}$$

Berdasarkan persamaan (3.9) maka disusunlah persamaan matriks turbin angin dalam bentuk berikut :

$$\begin{bmatrix} \dot{T}_g \\ \dot{\omega}_g \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} T_g \\ \omega_r \\ \omega_g \\ \theta_\Delta \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_a \\ T_{g,ref} \end{bmatrix}$$

dengan,

$$y(k) = \begin{bmatrix} \dot{T}_g \\ \dot{\omega}_g \end{bmatrix}; C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Selanjutnya parameter-parameter turbin angin pada Tabel 3.2 dimasukkan ke dalam matriks tersebut. Sehingga nilai matriks ruang keadaan untuk perancangan *observer* diperoleh dari pemodelan sistem turbin angin adalah sebagai berikut.

$$A = \begin{bmatrix} 0.9900 & 0 & 0 & 0 \\ -6.425 \times 10^{-8} & 0.9998 & 0.0002 & -0.0093 \\ -0.0007 & 0.0011 & 0.9989 & 0.0586 \\ 3.636 \times 10^{-7} & 0.0009 & -0.0009 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0.0099 \\ 0.0001 & -2.134 \times 10^{10} \\ 6.446 \times 10^{-8} & -3.636 \times 10^{-6} \\ 5.785 \times 10^{-8} & 1.213 \times 10^{-9} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$Fs = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Persamaan ruang keadaan yang mengandung kesalahan sensor dituliskan sebagai berikut :

$$x(k+1) = Ax(k) + Bu(k) \quad (3.10)$$

$$y(k) = Cx(k) + Fs \cdot f_s(k) \quad (3.11)$$

Dengan, $x(k) \in R^n$, $u(k) \in R^m$, $y(k) \in R^p$, $f_s(k) \in R^n$ berturut-turut adalah keadaan(*state*), masukan, keluaran, dan kesalahan sensor. Sedangkan A, B, C, F_s adalah matriks keadaan, masukan, keluaran dan kesalahan sensor.

Kemudian *state* baru $z(k+1)$ ditambahkan untuk memindahkan kesalahan sensor dari persamaan keluaran ke persamaan keadaan

$$z(k+1) = A_z T_s (y(k) - z(k)) + z(k) \quad (3.12)$$

dimana persamaan (3.11) disubstitusi ke dalam persamaan (3.12), sehingga diperoleh persamaan :

$$z(k+1) = -A_z T_s z(k) + A_z T_s (Cx(k) + Fs \cdot f_s(k)) + z(k) \quad (3.13)$$

persamaan (3.10), (3.13) diubah ke dalam bentuk persamaan yang lebih kompak menjadi sebagai berikut.

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ z(k+1) \end{bmatrix} &= \begin{bmatrix} A & 0 \\ A_z C & -A_z T_s + I \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) \\ &\quad + \begin{bmatrix} 0 & 0 \\ 0 & A_z T_s F_s \end{bmatrix} \begin{bmatrix} f_a(k) \\ f_s(k) \end{bmatrix} \end{aligned} \quad (3.14)$$

$$y(k) = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix} \quad (3.15)$$

dengan nilai :

$$A_z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, T_s = 0.1$$

Kemudian bentuk ruang keadaan dirubah kedalam bentuk persamaan keadaan dan persamaan keluaran sistem yang diperluas (mengandung kesalahan) sebagai berikut

$$\bar{x}(k+1) = \bar{A}.\bar{x}(k) + \bar{B}.u(k) + \bar{F}\bar{f}(k) \quad (3.16)$$

$$z(k) = \bar{y}(k) = \bar{C}\bar{x}(k) \quad (3.17)$$

dengan demikian diperoleh nilai matriks sebagai berikut.

$$\bar{A} = \begin{bmatrix} 0.9900 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9998 & 0.0002 & -0.0093 & 0 & 0 \\ -0.0007 & 0.0011 & 0.9989 & 0.0586 & 0 & 0 \\ 0 & 0.0010 & -0.0010 & 1.0000 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\bar{B} = \begin{bmatrix} 0 & 0.0100 \\ 0.0001 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\bar{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix};$$

$$\bar{F} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Setelah nilai-nilai matriks didapatkan, persamaan keadaan dan persamaan keluaran yang mengandung sensor digunakan untuk merancang persamaan *observer* yang dituliskan:

$$\hat{\bar{x}}(k+1) = \bar{A}.\hat{\bar{x}} + \bar{B}.u(k) + \bar{F}\hat{\bar{f}}(k) + K_e(\bar{y}(k) - \hat{\bar{y}}(k)) \quad (3.18)$$

$$\hat{\bar{y}}(k) = \bar{C}\hat{\bar{x}}(k) \quad (3.19)$$

Kemudian ditambahkan *state* yang mengandung *gain* untuk mengestimasi kesalahan sensor (L),

$$\hat{\bar{f}}(k+1) = T_s L_e(k) + \hat{\bar{f}}(k) \quad (3.20)$$

Sehingga dari persamaan (3.18) dan (3.19) jika diubah kedalam bentuk yang lebih kompak, menjadi sebagai berikut.

$$\begin{bmatrix} \hat{\tilde{x}}(k+1) \\ \hat{\tilde{f}}(k+1) \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{F} \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{\tilde{x}}(k) \\ \hat{\tilde{f}}(k) \end{bmatrix} + \begin{bmatrix} \bar{B} \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} K \\ LT_s \end{bmatrix} e(k) \quad (3.21)$$

$$\hat{\tilde{y}}(k) = [\bar{C} \quad 0] \begin{bmatrix} \hat{\tilde{x}}(k) \\ \hat{\tilde{f}}(k) \end{bmatrix} \quad (3.22)$$

dari bentuk (3.21) dan (3.22) dapat dirubah menjadi persamaan keadaan dan persamaan keluaran *observer*, sebagai berikut :

$$\hat{\tilde{x}}(k+1) = \tilde{A} \cdot \hat{\tilde{x}}(k) + \tilde{B} \cdot u(k) + \tilde{K}_e(\tilde{y}(k) - \hat{\tilde{y}}(k)) \quad (3.23)$$

$$\hat{\tilde{y}}(k) = \tilde{C} \cdot \hat{\tilde{x}}(k) \quad (3.24)$$

dengan memasukan nilai matriks $\bar{A}, \bar{B}, \bar{C}, \bar{F}, I, Ts$ maka, diperoleh nilai matriks sebagai berikut :

$$\tilde{A} = \begin{bmatrix} 0.9900 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9998 & 0.0002 & -0.0093 & 0 & 0 & 0 \\ -0.0007 & 0.0011 & 0.9989 & 0.0586 & 0 & 0 & 0 \\ 0 & 0.0010 & -0.0010 & 1.0000 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} 0 & 0.0100 \\ 0.0001 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\tilde{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Sehingga dari persamaan *observer* yang telah diperoleh, untuk menentukan nilai *error observer* sebagai berikut.

$$e(k+1) = \tilde{x}(k+1) - \hat{\tilde{x}}(k+1) \quad (3.25)$$

$$\begin{aligned} e(k+1) &= (\tilde{A}\tilde{x}(k) + \tilde{B}u(k)) - (\tilde{A} \cdot \hat{\tilde{x}}(k) + \tilde{B} \cdot u(k) \\ &\quad + \tilde{K}_e(\tilde{y}(k) - \hat{\tilde{y}}(k))) \end{aligned} \quad (3.26)$$

$$e(k+1) = \tilde{A}\tilde{x}(k) - \tilde{A} \cdot \hat{\tilde{x}}(k) - \tilde{K}_e(\tilde{C} \cdot \tilde{x}(k) - \tilde{C} \cdot \hat{\tilde{x}}(k)) \quad (3.27)$$

$$e(k+1) = (\tilde{A} - \tilde{K}_e\tilde{C})(\tilde{x}(k) - \hat{\tilde{x}}(k)) \quad (3.28)$$

dengan,

$$e(k) = \tilde{x}(k) - \hat{\tilde{x}}(k) \quad (3.29)$$

sehingga,

$$e(k+1) = (\tilde{A} - \tilde{K}_e \tilde{C})e(k) \quad (3.30)$$

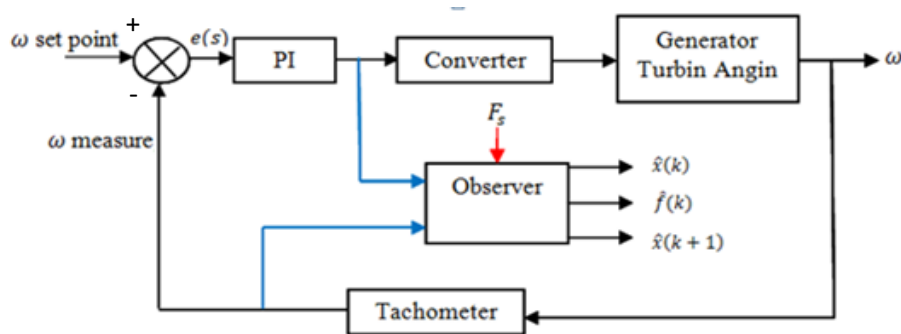
Langkah selanjutnya, Untuk memperoleh nilai *gain observer* (\tilde{K}_e) digunakan metode *pole-placement*. Nilai *pole* yang diinginkan ditentukan dengan metode *trial-and-error*, dengan berdasarkan ilmu kestabilan berdasarkan letak *pole*, diketahui bahwa sebuah sistem diskrit didefinisikan stabil jikalau seluruh akar-akar karakteristik bagian riil atau imajiner bernilai tidak lebih dari 1 atau -1. Sehingga dengan metode *trial-and-error* diperoleh 7 pole yaitu 0,92; 0,91; 0,9388763; 0,94; 0,919; 0,9911 dan 0,99911.

Sehingga diperoleh matriks K_e sebagai berikut :

$K_e =$

$$\begin{bmatrix} 0.0004 & 0.6546 & 0.6499 & 0.0006 & 0.0001 & -0.0000 & -0.6548 \\ -0.000 & -5.0609 & -5.0286 & -0.0054 & -0.0000 & 0.0002 & 5.0628 \end{bmatrix}$$

Setelah dilakukan perancangan *observer* secara simulasi menggunakan perangkat lunak MATLAB R2014a, dan *observer* dapat mengestimasi kesalahan dengan benar. Gambar 3.3 berikut ini adalah gambar diagram blok sistem pengendalian yang telah diintegrasikan dengan *observer*.



Gambar 3.3 Diagram blok sistem pengendalian yang telah terintegrasikan dengan *observer*

3.4 Pengujian Algoritma *Observer*

Pengujian algoritma *observer* bertujuan untuk memastikan apakah *observer* berupa hasil estimasi kesalahan sensor ($\hat{f}(k)$) yang dibuat telah

sesuai dengan kesalahan riil yang telah dirancang. Pengujian dilakukan dengan menggunakan sinyal kesalahan berupa step.

3.5 Perancangan Algoritma Prediksi Kesalahan

Setelah perancangan algoritma *observer*, langkah berikutnya perancangan algoritma prediksi kesalahan yang mengacu pada (Xu et al., 2009) yaitu algoritma *exponential smooting*. Estimasi kesalahan yang diperhalus dapat dihitung dengan persamaan sebagai berikut:

$$u_n = a\hat{\alpha}_n + (1 - a)(u_{n-1} + s_{n-1}) \quad (3.31)$$

$$s_n = b(u_n - u_{n-1}) + (1 - b)s_{n-1} \quad (3.32)$$

Misalkan u_n adalah hasil estimasi yang diperhalus dan s_n adalah faktor pertumbuhan linear yang diestimasi. Diberikan hasil estimasi kesalahan $\hat{\alpha}_n (n = 1, 2, \dots)$ yang diperoleh dari *observer*. Dimana $0 < a < 1$ & $0 < b < 1$ adalah parameter *Holt's method* (Xu et al., 2009). Kondisi awal untuk *Holt's method* sebagai berikut:

$$u_1 = \hat{\alpha}_1 = \hat{f}(k) \quad (3.33)$$

$$u_2 = \hat{\alpha}_2 = \hat{f}(k + 1) \quad (3.34)$$

$$s_2 = u_2 - u_1 \quad (3.35)$$

Setelah menghitung estimasi kesalahan yang diperhalus dan faktor pertumbuhan pada saat nT prediksi kesalahan kemudian dapat dihitung menggunakan algoritma berikut (Xu et al., 2009):

$$\bar{\alpha}_{n+l} = u_n + s_n lT \quad (3.36)$$

dengan

l : step prediksi untuk prediksi kesalahan

T : time interval

Untuk $n = 1$ atau $n = 2$ menggunakan persamaan (3.33) atau (3.34) untuk $n > 2$ menggunakan persamaan (3.31).

3.6 Pengujian Algoritma Prediksi Kesalahan Sensor

Pengujian algoritma prediksi kesalahan sensor dilakukan untuk memastikan apakah algoritma *observer* dapat diterapkan pada algoritma prediksi kesalahan sensor yang mengacu algoritma *exponential smoothing*. Hasil estimasi kesalahan menggunakan *observer* digunakan untuk menghitung estimasi kesalahan diperhalus (u_n) untuk $n \geq 2$ dan menghitung faktor pertumbuhan linear (s_n) untuk $n \geq 2$. Dimana (u_n) dan (s_n) mengandung parameter *Holt's method* a dan b dimana $0 < a < 1$ & $0 < b < 1$. Pengujian ini dilakukan dengan memberikan variasi nilai T (*time interval*) 1 detik, 5 detik, 10 detik dan 50 detik. Setiap nilai *time interval* diberikan variasi jumlah prediksi kesalahan $\hat{f}(k+1)$, $\hat{f}(k+5)$ dan $\hat{f}(k+10)$. Pengujian dilakukan dengan menggunakan sinyal kesalahan berupa *ramp*.

3.7 Perancangan Algoritma Prediksi Keandalan *Real-Time*

Langkah terakhir yaitu perancangan algoritma prediksi keandalan *real-time* yang mengacu pada (Xu et al., 2009).

- 1) Inisialisasi partikel dimulai dari $l = 1$.

$$\bar{X}_{n+l-1|n}^{(i)} = \bar{X}_{n+0|n}^{(i)} = \hat{X}_{n|n}^{(i)}$$

- 2) Estimasi kesalahan menggunakan algoritma *observer* yang akan menghasilkan *state* estimasi mengacu pada (Indriawati et al., 2015).

$$u_1 = \hat{\alpha}_1 = \hat{f}(k)$$

$$u_2 = \hat{\alpha}_2 = \hat{f}(k+1)$$

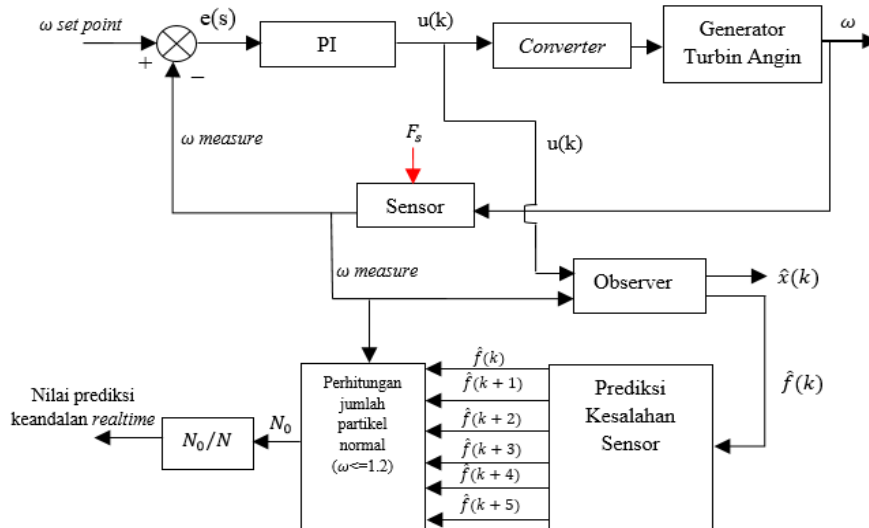
- 3) Prediksi kesalahan menggunakan algoritma *exponential smoothing* pada persamaan (3.30), (3.31), dan (3.35).

- 4) Menghitung partikel baru menggunakan persamaan :

$$\bar{X}_{n+l-1|n}^{(i)} = f(\bar{X}_{n+l-1|n}^{(i)}) + \Gamma \tilde{\alpha}_{n+l} + W_{n+l}^{(i)} \text{ dimana } W_{n+l}^{(i)} \text{ adalah realisasi dari } w_{n+l} \text{ untuk partikel ke-} i.$$

- 5) Pemilihan partikel jika $l < r$, maka $l = l + 1$, dan kembali ke step 2, jika sebaliknya lanjut pada step 6.

- 6) Menghitung jumlah partikel dari himpunan $\{\bar{X}_{n+r|n}^{(i)}\}(i = 1, \dots, N)$ yang jatuh pada \mathcal{M}_0 (area yang diperbolehkan) dan dinotasikan oleh N_0 . Maka sistem prediksi keandalan selama $(nT, (n+r)T]$ dihitung oleh N_0/N .



Gambar 3.4 Diagram blok sistem prediksi keandalan *real-time*

Gambar 3.4 merupakan diagram blok sistem prediksi keandalan *real-time* yang telah diintegrasikan dengan sistem pengendalian kecepatan generator turbin angin, *observer*, dan prediksi kesalahan sensor. $\hat{f}(k)$ merupakan nilai estimasi kesalahan dari *observer* kemudian digunakan dalam prediksi kesalahan sensor. Hasil prediksi kesalahan sensor digunakan matriks untuk mengambil *state* prediksi kesalahan sensor. Selanjutnya dibandingkan nilai *state* prediksi kesalahan sensor dengan nilai maksimum kecepatan sebesar 1.2 pu, jika *state* prediksi kesalahan sensor melebihi nilai 1.2 pu maka sistem dikatakan gagal. Lalu pada blok N_0/N merupakan hasil jumlah partikel yang memasuki area yang diperbolehkan \mathcal{M}_0 yang dinotasikan N_0 (kecepatan tidak melebihi ≤ 1.2 pu) kemudian dibandingkan jumlah tersebut dengan jumlah seluruh partikel (N).

3.8 Pengujian Algoritma Prediksi Keandalan *Real-time*

Pengujian algoritma prediksi keandalan *real-time* dilakukan untuk menentukan nilai parameter desain yang mempengaruhi prediksi keandalan *real-time* pada sistem pengendalian kecepatan generator turbin angin. Hasil prediksi kesalahan sensor digunakan dalam memprediksi keandalan *real-time*. Pengujian

ini dilakukan dengan memberikan variasi nilai T (*time interval*) 1 detik, 5 detik, 10 detik dan 50 detik. Setiap nilai *time interval* diberikan variasi jumlah prediksi satu prediksi, lima prediksi dan sepuluh prediksi. Pengujian hasil prediksi keandalan *real-time* dilakukan dengan prosedur sebagai berikut:

- a. prediksi kesalahan sensor yang dihasilkan melalui algoritma *exponential smoothing* selanjutnya digunakan matriks untuk mengambil *state* prediksi kesalahan sensor.
- b. dibandingkan nilai *state* prediksi kesalahan sensor dengan nilai maksimum kecepatan sebesar 1.2 pu, jika *state* prediksi kesalahan sensor melebihi nilai 1.2 pu maka sistem dikatakan gagal.
- c. Lalu pada N_0/N merupakan hasil jumlah partikel yang memasuki area yang diperbolehkan \mathcal{M}_0 yang dinotasikan N_0 (kecepatan tidak melebihi ≤ 1.2 pu) kemudian dibandingkan jumlah tersebut dengan jumlah seluruh partikel (N).

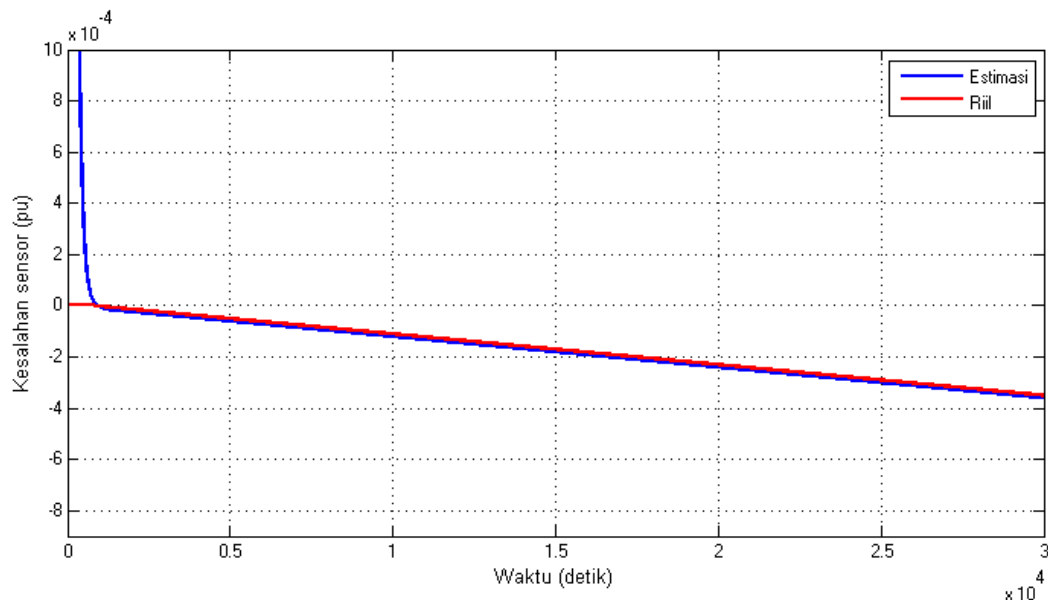
BAB 4

HASIL DAN PEMBAHASAN

Bab ini membahas mengenai hasil simulasi dari sistem pengendalian kecepatan generator turbin angin menggunakan MATLAB 2014a. Terdapat tiga pengujian untuk simulasi yang dilakukan yaitu: pengujian *observer*, pengujian algoritma prediksi kesalahan sensor dan pengujian algoritma prediksi keandalan *real-time*.

4.1. Hasil Uji *Observer*

Uji *observer* dilakukan dengan tujuan untuk mengetahui kemampuan *observer* dalam mengestimasi kesalahan sensor. Kesalahan sensor disimulasikan sebagai kesalahan sensitivitas sebesar α pada tachometer sehingga hasil pengukuran yang salah adalah $\alpha \times \omega$ (pu). Kesalahan sensor yang disimulasikan berupa sinyal *ramp* dengan *slope* $10^{-8} t$ dan kesalahan diberikan pada detik ke-800. Garis berwarna merah muda menunjukkan hasil estimasi kesalahan sensor menggunakan *observer* dan garis berwarna merah menunjukkan hasil riil kesalahan sensor.



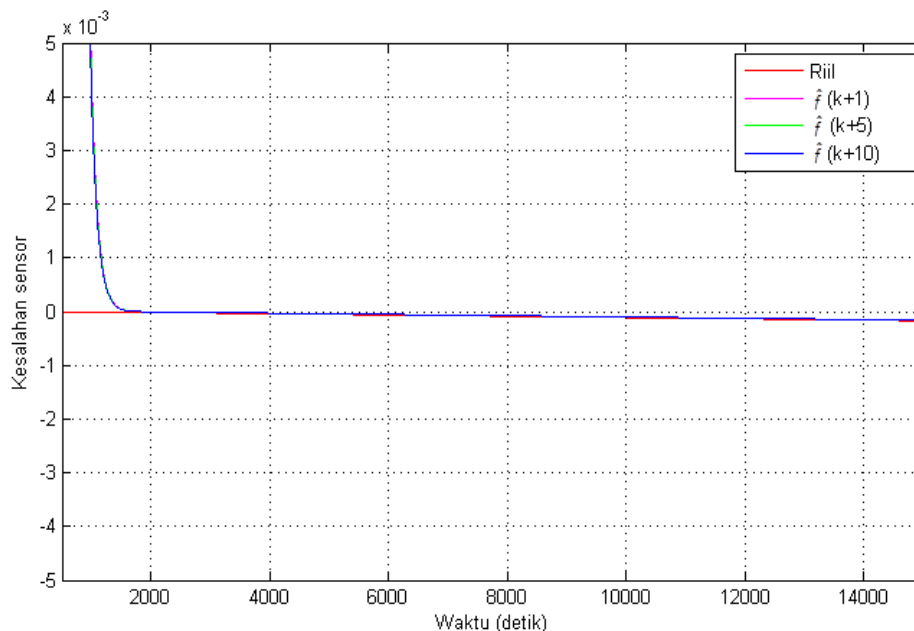
Gambar 4.1 Grafik hasil estimasi kesalahan sensor

Gambar 4.1 menunjukkan *observer* mampu mengestimasi kesalahan sensor dengan baik. Kontroler PI terbukti mampu membuat respon sistem sesuai

kriteria yang diinginkan, seperti yang ditunjukkan pada saat terjadi perubahan nilai estimasi baik di awal simulasi maupun pada detik ke-800, yaitu tanpa *overshoot* dan *settling time* sekitar 700 detik. *Observer* tetap dapat mempertahankan kecepatan dan ketepatan respon hasil estimasi. Berdasarkan Gambar 4.1, tampak bahwa hasil estimasi kesalahan sensor mengikuti dinamika kesalahan riil yang terjadi.

4.2. Hasil Uji Algoritma Prediksi Kesalahan Sensor

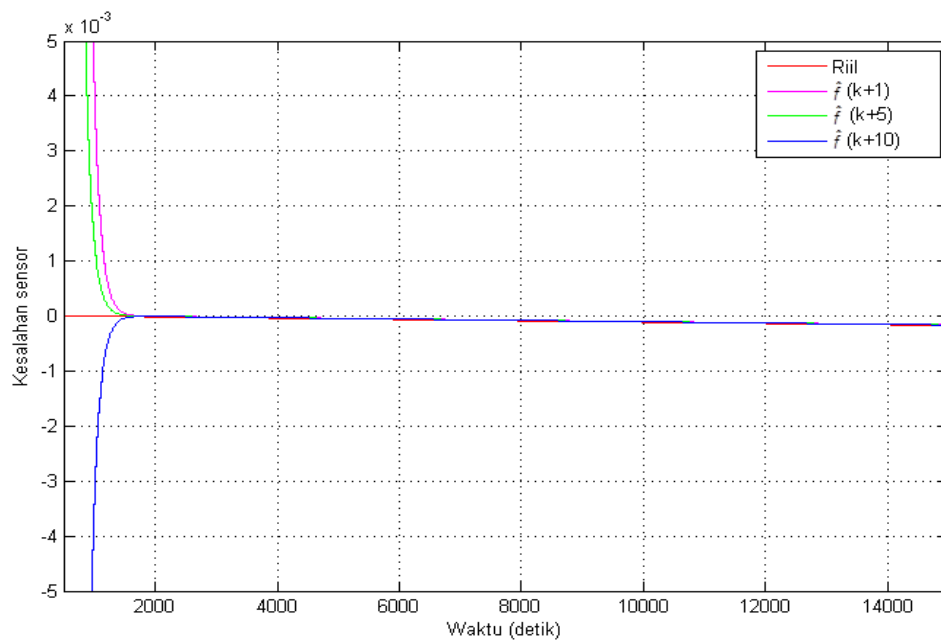
Prediksi kesalahan sensor diterapkan menggunakan hasil estimasi kesalahan sensor dari *observer*. Algoritma prediksi kesalahan sensor mengacu pada algoritma *exponential smoothing*. Parameter *Holt's method* pada algoritma *exponential smoothing* diberikan sebesar $a = 0.9$, dan $b = 0.9$. Pengujian ini dilakukan dengan memberikan variasi nilai T (*time interval*). Nilai *time interval* dipilih dengan berbagai variasi nilai untuk mengetahui prediksi kesalahan yang dihasilkan. Setiap nilai *time interval* diberikan variasi jumlah prediksi. Hubungan kesalahan sensor terhadap waktu ditampilkan pada Gambar 4.2-4.5.



Gambar 4.2 Grafik hasil uji prediksi kesalahan sensor dengan *time interval* 1 detik

Gambar 4.2. menunjukkan algoritma *exponential smoothing* dapat memprediksi kesalahan sensor dengan benar. Kontroler PI terbukti mampu

membuat respon sistem sesuai kriteria yang diinginkan, seperti yang ditunjukkan pada saat terjadi perubahan nilai prediksi baik di awal simulasi maupun pada detik ke-800, yaitu tanpa *overshoot* dan *settling time* sekitar 626 detik untuk $\hat{f}(k+1)$, 623 detik untuk $\hat{f}(k+5)$, dan 617 detik untuk $\hat{f}(k+10)$. Performansi prediksi kesalahan sensor $\hat{f}(k+10)$ mempunyai performansi terbaik jika dibandingkan dengan jumlah prediksi kesalahan sensor lainnya. Hal ini didasarkan pada nilai *settling time* yang lebih cepat jika dibandingkan dengan *settling time* dari variasi jumlah prediksi kesalahan sensor lainnya.

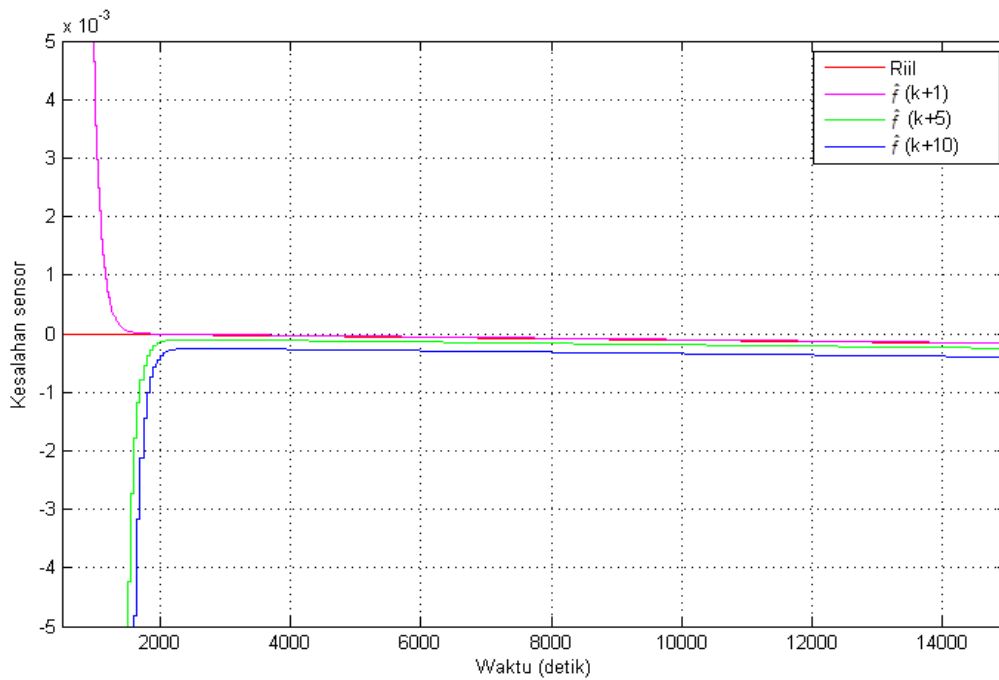


Gambar 4.3 Grafik hasil uji prediksi kesalahan sensor dengan *time interval* 5 detik

Hasil uji prediksi kesalahan sensor dengan *time interval* 5 detik ditunjukkan Gambar 4.3. Hasil prediksi kesalahan sensor dengan *time interval* 5 detik untuk $\hat{f}(k+10)$ mengalami penyimpangan respon jika dibandingkan dengan $\hat{f}(k+1)$ dan $\hat{f}(k+5)$. Respon yang dihasilkan oleh $\hat{f}(k+10)$ mengalami kenaikan dari -5×10^{-3} sampai 0 pada rentang waktu 900-1420 detik, sedangkan respon yang dihasilkan oleh $\hat{f}(k+1)$ dan $\hat{f}(k+5)$ mengalami penurunan dari 5×10^{-3} sampai 0 pada rentang waktu 900-1420 detik. Terjadinya penyimpangan pada *time interval* 5 detik dikarenakan algoritma *exponential*

smoothing tidak mampu memprediksi dengan benar pada $\hat{f}(k+10)$ (Xu et al., 2008).

Kontroler PI terbukti mampu membuat respon sistem sesuai kriteria yang diinginkan, seperti yang ditunjukkan pada saat terjadi perubahan nilai prediksi baik di awal simulasi maupun pada detik ke-800, yaitu tanpa *overshoot* dan *settling time* sekitar 630 detik untuk $\hat{f}(k+1)$, 500 detik untuk $\hat{f}(k+5)$, dan 600 detik untuk $\hat{f}(k+10)$. Performansi prediksi kesalahan sensor $\hat{f}(k+5)$ mempunyai performansi terbaik jika dibandingkan dengan jumlah prediksi kesalahan sensor lainnya. Hal ini didasarkan pada nilai *settling time* yang lebih cepat jika dibandingkan dengan *settling time* dari variasi jumlah prediksi kesalahan sensor lainnya.

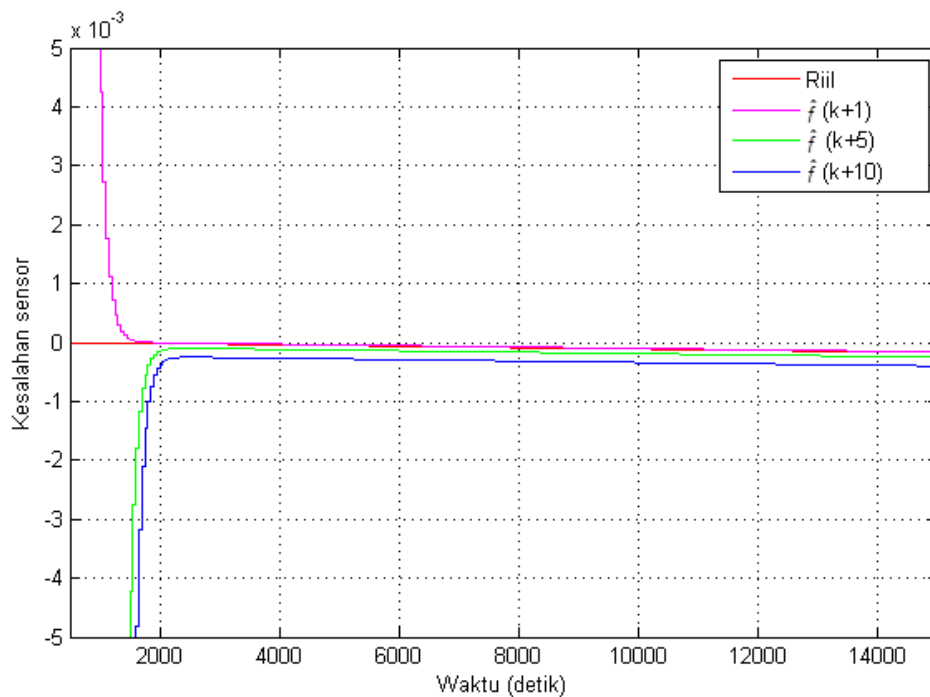


Gambar 4.4 Grafik hasil uji prediksi kesalahan sensor dengan *time interval* 10 detik

Hasil uji prediksi kesalahan sensor dengan *time interval* 10 detik ditunjukkan Gambar 4.4. Hasil prediksi kesalahan sensor dengan *time interval* 10 detik untuk $\hat{f}(k+5)$ dan $\hat{f}(k+10)$ mengalami penyimpangan respon jika dibandingkan dengan $\hat{f}(k+1)$. Respon yang dihasilkan oleh $\hat{f}(k+5)$ dan $\hat{f}(k+10)$ mengalami kenaikan dari -5×10^{-3} sampai 0 pada rentang waktu 1500-

1900 detik, sedangkan respon yang dihasilkan oleh $\hat{f}(k+1)$ mengalami penurunan dari 5×10^{-3} sampai 0 pada rentang waktu 950-1430 detik. Terjadinya penyimpangan pada *time interval* 10 detik dikarenakan algoritma *exponential smoothing* tidak mampu memprediksi dengan benar pada $\hat{f}(k+10)$ (Xu et al., 2008).

Kontroler PI terbukti mampu membuat respon sistem sesuai kriteria yang diinginkan meskipun menghasilkan *error steady state* sebesar $1,25 \times 10^{-4}$ pada $\hat{f}(k+5)$ dan $3,75 \times 10^{-4}$ pada $\hat{f}(k+10)$. Respon yang dihasilkan tanpa *overshoot* dan *settling time* sekitar 630 detik untuk $\hat{f}(k+1)$, 670 detik untuk $\hat{f}(k+5)$, dan 690 detik untuk $\hat{f}(k+10)$. Performansi prediksi kesalahan sensor $\hat{f}(k+1)$ mempunyai performansi terbaik jika dibandingkan dengan jumlah prediksi kesalahan sensor lainnya. Hal ini didasarkan pada nilai *settling time* yang lebih cepat dibandingkan dengan jumlah prediksi kesalahan sensor lainnya.



Gambar 4.5 Grafik hasil uji prediksi kesalahan sensor dengan *time interval* 50 detik

Hasil uji prediksi kesalahan sensor dengan *time interval* 50 detik ditunjukkan Gambar 4.5. Hasil prediksi kesalahan sensor dengan *time interval* 50 detik untuk $\hat{f}(k+5)$ dan $\hat{f}(k+10)$ mengalami penyimpangan respon jika

dibandingkan dengan $\hat{f}(k+1)$. Respon yang dihasilkan oleh $\hat{f}(k+5)$ dan $\hat{f}(k+10)$ mengalami kenaikan dari -5×10^{-3} sampai 0 pada rentang waktu 1500-1900 detik, sedangkan respon yang dihasilkan oleh $\hat{f}(k+1)$ mengalami penurunan dari 5×10^{-3} sampai 0 pada rentang waktu 950-1430 detik. Terjadinya penyimpangan pada *time interval* 10 detik dikarenakan algoritma *exponential smoothing* tidak mampu memprediksi dengan benar pada $\hat{f}(k+10)$ (Xu et al., 2008).

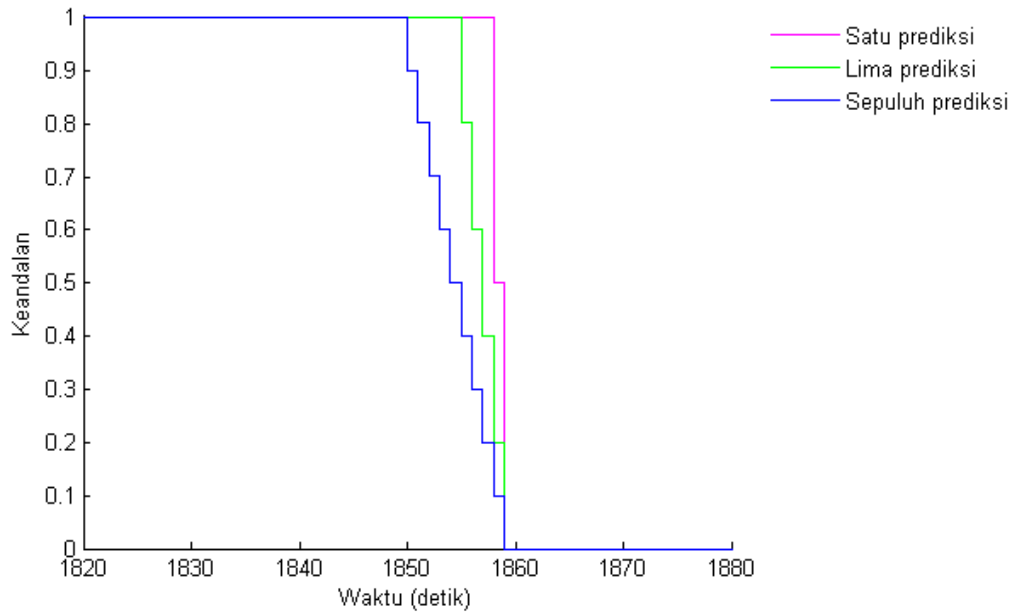
Kontroler PI terbukti mampu membuat respon sistem sesuai kriteria yang diinginkan meskipun menghasilkan *error steady state* sebesar $1,25 \times 10^{-4}$ pada $\hat{f}(k+5)$ dan $3,75 \times 10^{-4}$ pada $\hat{f}(k+10)$. Respon yang dihasilkan tanpa *overshoot* dan *settling time* sekitar 650 detik untuk $\hat{f}(k+1)$, 1100 detik untuk $\hat{f}(k+5)$, dan 1200 detik untuk $\hat{f}(k+10)$. Performansi prediksi kesalahan sensor $\hat{f}(k+1)$ mempunyai performansi terbaik jika dibandingkan dengan jumlah prediksi kesalahan sensor lainnya. Hal ini didasarkan pada nilai *settling time* yang lebih cepat dibandingkan dengan jumlah prediksi kesalahan sensor lainnya.

Gambar 4.2-4.5 menunjukkan bahwa variasi nilai *time interval* yang diberikan mempengaruhi prediksi kesalahan sensor yang dihasilkan. Untuk *time interval* di atas 10 detik algoritma *exponential smoothing* yang digunakan pada prediksi kesalahan sensor tidak mampu memprediksi dengan benar, yaitu terjadi *error steady state* sebesar $1,25 \times 10^{-4}$ untuk *time interval* 10 detik dan $3,75 \times 10^{-4}$ untuk *time interval* 50 detik. Sedangkan untuk *time interval* 1 detik dan 5 detik, algoritma *exponential smoothing* dapat mempertahankan kecepatan dan ketepatan respon hasil prediksi.

4.3. Hasil Uji Algoritma Prediksi Keandalan *Real-time*

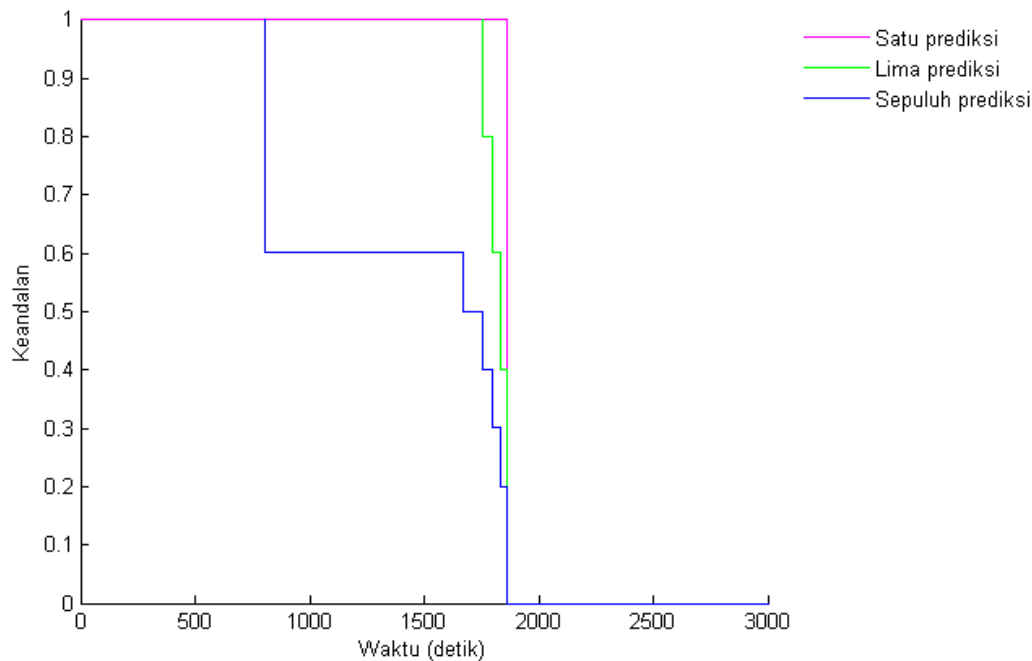
Prediksi keandalan *real-time* diterapkan menggunakan hasil prediksi kesalahan sensor. Hasil uji algoritma prediksi keandalan *real-time* bertujuan untuk menentukan nilai parameter desain yang mempengaruhi prediksi keandalan *real-time* pada sistem pengendalian kecepatan generator turbin angin berdasarkan prediksi kesalahan sensor. Pengujian ini dilakukan dengan memberikan variasi nilai T (*time interval*). Nilai *time interval* dipilih dengan berbagai variasi nilai untuk mengetahui nilai prediksi keandalan *real-time* yang dihasilkan. Setiap nilai

time interval diberikan variasi jumlah prediksi. Hubungan prediksi keandalan *real-time* terhadap waktu ditampilkan pada Gambar 4.6-4.9.



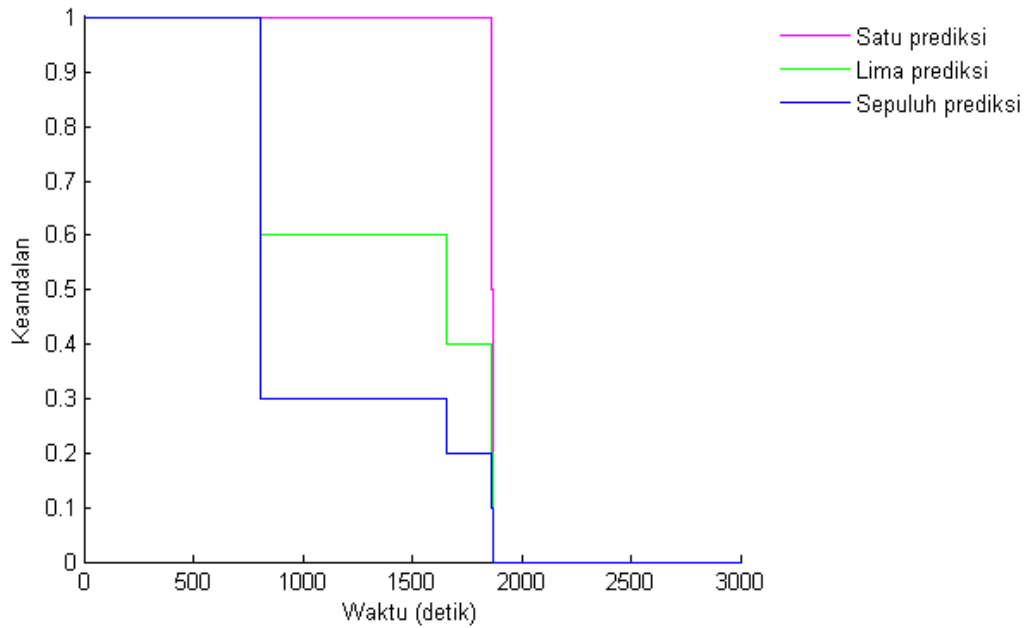
Gambar 4.6 Grafik hasil uji prediksi keandalan *real-time* dengan *time interval* 1 detik

Hasil uji prediksi keandalan *real-time* dengan *time interval* 1 detik ditunjukkan Gambar 4.6. Prediksi kesalahan sensor mempengaruhi prediksi keandalan *real-time* yang dihasilkan. Waktu kegagalan riil ketika kecepatan putar lebih besar dari 1.2 pu pada detik ke-1859 untuk setiap jumlah prediksi, sedangkan kecepatan putar dibawah 0.5 pu keandalan mulai turun pada detik ke-1859 untuk satu prediksi, detik ke-1855 untuk lima prediksi dan detik ke-1850 untuk sepuluh prediksi. Prediksi keandalan *real-time* dengan *time interval* 1 detik untuk sepuluh prediksi mempunyai performansi terbaik jika dibandingkan dengan jumlah prediksi lainnya. Hal ini didasarkan pada waktu keandalan mulai turun untuk sepuluh prediksi lebih teliti, tampak bahwa pada detik ke 1850 sudah terdeteksi terjadi kesalahan sensor. Hasil tersebut sesuai dengan penelitian sebelumnya yang diaplikasikan pada *three-vessel water tank system* bahwa semakin besar jumlah prediksi maka tingkat prediksi keandalan akan semakin teliti (Xu et al., 2009).



Gambar 4.7 Grafik hasil uji prediksi keandalan *real-time* dengan *time interval* 5 detik

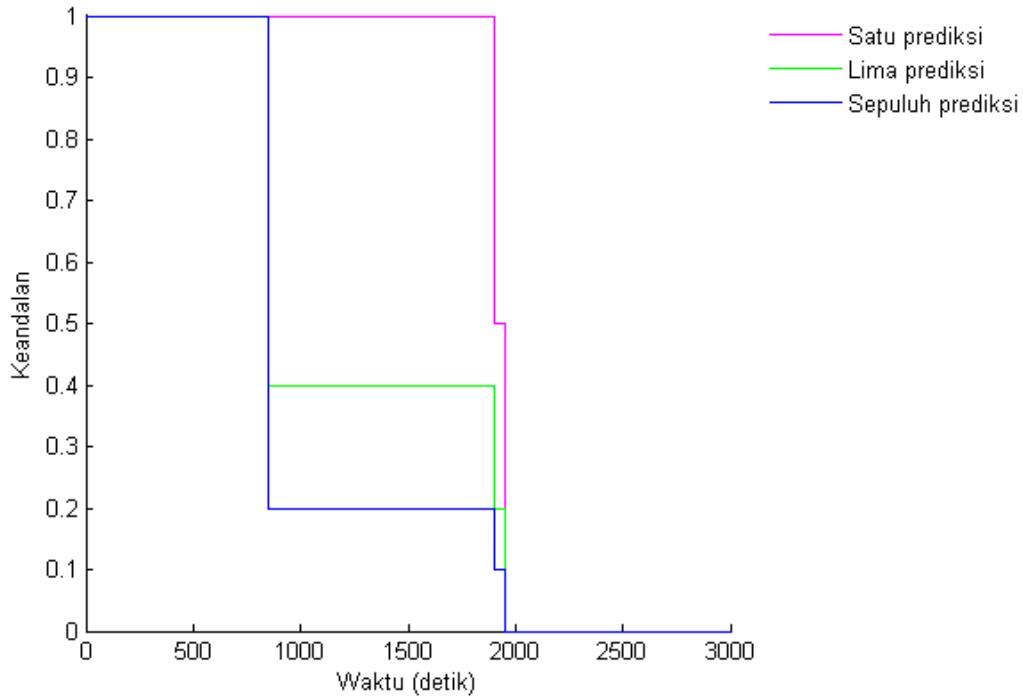
Hasil uji prediksi keandalan *real-time* dengan *time interval* 5 detik ditunjukkan Gambar 4.7. Prediksi kesalahan sensor $\hat{f}(k + 10)$ (lihat Gambar 4.3) mempengaruhi prediksi keandalan *real-time* sepuluh prediksi yang dihasilkan, terlihat bahwa keandalan untuk sepuluh prediksi lebih cepat mengalami penurunan keandalan. Waktu kegagalan riil ketika kecepatan putar lebih besar dari 1.2 pu pada detik ke-1865 untuk setiap jumlah prediksi, sedangkan kecepatan putar dibawah 0.5 pu keandalan mulai turun pada detik ke-1800 untuk satu prediksi, detik ke-1755 untuk lima prediksi dan detik ke-805 untuk sepuluh prediksi. Hasil tersebut sesuai dengan penelitian sebelum yang menyatakan bahwa nilai *time interval* mempengaruhi prediksi keandalan *real-time* yang dihasilkan (Xu et al., 2008). Prediksi keandalan *real-time* dengan *time interval* 5 detik untuk lima prediksi mempunyai performansi terbaik jika dibandingkan dengan sepuluh prediksi, karena pada sepuluh prediksi keandalan yang dihasilkan mengalami penurunan yang signifikan. Berdasarkan hasil sepuluh prediksi pada detik ke-805 terdeteksi adanya kesalahan sensor sebesar 5×10^{-8} , hal ini dikarenakan adanya penyimpangan pada prediksi kesalahan sensor $\hat{f}(k + 10)$ (lihat Gambar 4.3).



Gambar 4.8 Grafik hasil uji prediksi keandalan *real-time* dengan *time interval* 10 detik

Hasil uji prediksi keandalan *real-time* dengan *time interval* 10 detik ditunjukkan Gambar 4.8. Prediksi kesalahan sensor $\hat{f}(k+5)$ dan $\hat{f}(k+10)$ (lihat Gambar 4.4) mempengaruhi prediksi keandalan *real-time* sepuluh prediksi yang dihasilkan, terlihat bahwa keandalan untuk lima dan sepuluh prediksi lebih cepat mengalami penurunan keandalan. Waktu kegagalan riil ketika kecepatan putar lebih besar dari 1.2 pu pada detik ke-1865 untuk setiap jumlah prediksi, sedangkan kecepatan putar dibawah 0.5 pu keandalan mulai turun pada detik ke-1800 untuk satu prediksi, detik ke-1755 untuk lima prediksi dan detik ke-805 untuk sepuluh prediksi. Hasil tersebut sesuai dengan penelitian sebelum yang menyatakan bahwa nilai *time interval* mempengaruhi prediksi keandalan *real-time* yang dihasilkan (Xu et al., 2008). Prediksi keandalan *real-time* dengan *time interval* 10 detik untuk satu prediksi mempunyai performansi terbaik jika dibandingkan dengan sepuluh prediksi, karena pada lima dan sepuluh prediksi keandalan yang dihasilkan mengalami penurunan yang signifikan. Berdasarkan hasil lima dan sepuluh prediksi pada detik ke-810 terdeteksi adanya kesalahan sensor sebesar 10^{-7} , hal ini dikarenakan adanya penyimpangan pada prediksi

kesalahan sensor $\hat{f}(k + 5)$ dan $\hat{f}(k + 10)$ dan *error steady state* sebesar $1,25 \times 10^{-4}$ pada $\hat{f}(k + 5)$ dan $3,75 \times 10^{-4}$ pada $\hat{f}(k + 10)$ (lihat Gambar 4.4).



Gambar 4.9 Grafik hasil uji prediksi keandalan *real-time* dengan *time interval* 50 detik

Hasil uji prediksi keandalan *real-time* dengan *time interval* 50 detik ditunjukkan Gambar 4.9. Prediksi kesalahan sensor $\hat{f}(k + 5)$ dan $\hat{f}(k + 10)$ (lihat Gambar 4.5) mempengaruhi prediksi keandalan *real-time* sepuluh prediksi yang dihasilkan, terlihat bahwa keandalan untuk lima dan sepuluh prediksi lebih cepat mengalami penurunan keandalan. Waktu kegagalan riil ketika kecepatan putar lebih besar dari 1.2 pu pada detik ke-1865 untuk setiap jumlah prediksi, sedangkan kecepatan putar dibawah 0.5 pu keandalan mulai turun pada detik ke-1800 untuk satu prediksi, detik ke-1755 untuk lima prediksi dan detik ke-805 untuk sepuluh prediksi. Hasil tersebut sesuai dengan penelitian sebelum yang menyatakan bahwa nilai *time interval* mempengaruhi prediksi keandalan *real-time* yang dihasilkan (Xu et al., 2008). Prediksi keandalan *real-time* dengan *time interval* 50 detik untuk satu prediksi mempunyai performansi terbaik jika

dibandingkan dengan sepuluh prediksi, karena pada lima dan sepuluh prediksi keandalan yang dihasilkan mengalami penurunan yang signifikan. Berdasarkan hasil lima dan sepuluh prediksi pada detik ke-810 terdeteksi adanya kesalahan sensor sebesar 5×10^{-7} , hal ini dikarenakan adanya penyimpangan pada prediksi kesalahan sensor $\hat{f}(k+5)$ dan $\hat{f}(k+10)$ dan *error steady state* sebesar $1,25 \times 10^{-4}$ pada $\hat{f}(k+5)$ dan $3,75 \times 10^{-4}$ pada $\hat{f}(k+10)$ (lihat Gambar 4.5).

Gambar 4.6-4.9 menunjukkan bahwa semakin kecil nilai *time interval* maka prediksi keandalan *real-time* yang dihasilkan semakin tepat (Xu et al., 2008). Sedangkan semakin besar jumlah prediksi maka prediksi keandalan *real-time* yang dihasilkan semakin teliti (Xu et al., 2009). Apabila Gambar 4.6-4.9 dinyatakan dalam bentuk tabel, maka dapat dinyatakan sebagai Tabel 4.1 dan tabel prediksi keandalan *real-time* dengan *time interval* 1 detik sampai 50 detik ditampilkan pada Lampiran C.

Tabel 4.1 menunjukkan terdapat perbedaan nilai keandalan *real-time* dengan berbagai variasi *time interval* dan jumlah prediksi. *Time interval* sebesar 1 detik dengan jumlah prediksi sebesar 10 menghasilkan prediksi keandalan *real-time* terbaik. Hal ini didasarkan pada ketepatan dan ketelitian yang lebih baik dibandingkan dengan *time interval* dan jumlah prediksi lainnya. Prediksi keandalan *real-time* dengan *time interval* sebesar 5 detik dengan jumlah prediksi sebesar 1 dan 5 menghasilkan prediksi keandalan *real-time* lebih baik jika dibandingkan dengan *time interval* sebesar 10 dan 50 detik. Hal ini terlihat pada prediksi kesalahan sensor (Gambar 4.4 dan 4.5). Nilai *time interval* dan jumlah prediksi dipilih dengan berbagai variasi nilai untuk mengetahui prediksi keandalan *real-time* yang dihasilkan. Prediksi keandalan *real-time* ini kedepannya diharapkan dapat memberikan informasi yang berguna untuk pemeliharaan prediktif sistem turbin angin.

Tabel 4.1 Hasil Uji Prediksi Keandalan Real-Time

<i>Time Interval</i> (detik)	Waktu (detik)	Keandalan	Waktu (detik)	Keandalan	Waktu (detik)	Keandalan
		Satu prediksi		Lima prediksi		Sepuluh prediksi

<i>Time Interval</i> (detik)	Waktu (detik)	Keandalan	Waktu (detik)	Keandalan	Waktu (detik)	Keandalan
		Satu prediksi		Lima prediksi		Sepuluh prediksi
1	0-1858	1	0-1855	1	0-1850	1
	1858-1859	5	1855-1856	0.8	1850-1851	0.9
	1859	0	1856-1857	0.6	1851-1852	0.8
			1857-1858	0.4	1852-1853	0.7
			1858-1859	0.2	1853-1854	0.6
			1859	0	1854-1855	0.5
					1855-1856	0.4
					1856-1857	0.3
					1857-1858	0.2
					1858-1859	0.1
					1859	0
5	0-1860	1	0-1755	1	0-800	1
	1860-1865	0.5	1755-1800	0.8	800-1670	0.6
	1865	0	1800-1835	0.6	1670-1755	0.5
			1835-1860	0.4	1775-1800	0.4
			1860-1865	0.2	1800-1835	0.3
			1865	0	1835-1860	0.2
					1860-1865	0.1
					1865	0
10	0-1860	1	0-810	1	0-810	1
	1860-1870	0.5	810-1660	0.6	810-1660	0.3
	1870	0	1660-1860	0.4	1660-1860	0.2
			1860-1870	0.2	1860-1870	0.1
			0	0	0	0
50	0-1900	1	0-850	1	0-850	1
	1900-1950	0.5	850-1900	0.4	850-1900	0.2
	1950	0	1900-1950	0.2	1900-1950	0.1
			1950	0	1950	0

BAB 5

KESIMPULAN

5.1. Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini adalah

- Algoritma *exponential smoothing* berhasil diterapkan pada sistem pengendalian kecepatan generator turbin angin dengan menggunakan *observer* berstruktur PI yang mampu memprediksi kesalahan sensor secara cepat dengan benar. *Observer* menghasilkan respon dengan waktu *settling time* 700 detik.
- Variasi nilai *time interval* yang diberikan mempengaruhi prediksi kesalahan sensor yang dihasilkan. Untuk *time interval* di atas 10 detik algoritma *exponential smoothing* yang digunakan pada prediksi kesalahan sensor tidak mampu memprediksi dengan benar, yaitu terjadi *error steady state* sebesar $1,25 \times 10^{-4}$ untuk *time interval* 10 detik dan $3,75 \times 10^{-4}$ untuk *time interval* 50 detik. Sedangkan untuk *time interval* 1 detik dan 5 detik, algoritma *exponential smoothing* dapat mempertahankan kecepatan dan ketepatan respon hasil prediksi.
- Prediksi keandalan *real-time* yang dihasilkan dipengaruhi oleh variasi *time interval*. Prediksi keandalan dengan *time interval* sebesar 1 detik lebih tepat jika dibandingkan variasi *time interval* sebesar 5 detik, 10 detik, dan 50 detik.
- Jumlah prediksi mempengaruhi ketelitian prediksi keandalan *real-time* yang dihasilkan, semakin besar jumlah prediksi maka semakin teliti. Dengan *time interval* 1 detik, jumlah prediksi sebesar 10 menghasilkan prediksi keandalan *real-time* lebih baik dibandingkan jumlah prediksi sebesar 1 dan 5

5.2. Saran

Pada penelitian selanjutnya dapat dikembangkan prediksi keandalan *real-time* berdasarkan prediksi kesalahan aktuator. Dapat diaplikasikan pada sistem yang memerlukan perawatan prediktif.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- Basilio, J. C. (2015). A laboratory for a first course in control systems.
- Bentley, J. P. (2005). *Principles of Measurement Systems* (Fourth Edi). England: Prentice Hall.
- Blischke W & Prabhakar M. (2000). *Reliability: Modeling, Prediction, and Optimization*. Canada: John Wiley & Sons, Inc.
- Brodgesell, A. (2003). *Tachometers and Angular Speed Detectors*.
- Bruno, E., Oussama, S., & Frans, K. (2003). *Springer Tracts in Advanced Robotics* (Vol. 1). Germany: Springer-Verlag Berlin Heidelberg.
- Chan, D., & Mo, J. (2017). Life Cycle Reliability and Maintenance Analyses of Wind Turbines. *Energy Procedia*, 110(December 2016), 328–333.
<https://doi.org/10.1016/j.egypro.2017.03.148>
- Charles-de-Gaulle-Str. (2010). World Wind Energy Report 2009. *Wind Energy*.
- Chinnam, R. B. (1999). On-line Reliability Estimation of Individual Components, Using Degradation Signals, 48(4).
- Chinnam, R. B., & Mohan, P. (2002). *Online reliability estimation of physical systems using neural networks and wavelets. International Journal of Smart Engineering System Design* (Vol. 4).
<https://doi.org/10.1080/10255810215021>
- Chou, J., & Tu, W. (2011). Failure analysis and risk management of a collapsed large wind turbine tower. *Engineering Failure Analysis*, 18(1), 295–313.
<https://doi.org/10.1016/j.engfailanal.2010.09.008>
- Dhillon, B. S. (2005). *Reliability, Quality, and Safety for Engineers*.
- Ebeling, C. (1997). *An Introduction To Reliability and Maintainability Engineering*. Boston: McGraw-Hill.
- Fraden, J. (2016). *Handbook of Modern Sensors*. Springer.
- Gebraeel, N. (2006). Sensory-Updated Residual Life Distributions for Components With Exponential, 3(4), 382–393.
- Grall, A., Dieulle, L., Bérenguer, C., & Roussignol, M. (2002). Continuous-time

- predictive-maintenance scheduling for a deteriorating system. *IEEE Transactions on Reliability*, 51(2), 141–150.
<https://doi.org/10.1109/TR.2002.1011518>
- H, Alwi. C, Edwards. & C, P. (2016). *Advances in Industrial Control*. London Dordrecht Heidelberg New York: Springer.
- Indriawati, K., Sebe, N., Agustinah, T., & Jazidie, A. (2015). Robust Fuzzy Observer-Based Fault Tolerant Tracking Control for Nonlinear Systems with Simultaneous Actuator and Sensor Faults: Application to a DC Series Motor Speed Drive. *International Review of Automatic Control (IREACO)*, 8(6), 375. <https://doi.org/10.15866/ireaco.v8i6.7505>
- Isermann, R. (2006). *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. New York, USA: Springer-Verlag.
- Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7), 1483–1510.
<https://doi.org/10.1016/j.ymssp.2005.09.012>
- Jha, A. R. (2011). *Wind Turbine Technology*. United States of America: CRC Press.
- Kharoufeh, J. P. (2003). Explicit results for wear processes in a Markovian environment, 31, 237–244.
- Kharoufeh, J. P., & Cox, S. M. (2005). Stochastic models for degradation-based reliability 1, 533–542. <https://doi.org/10.1080/07408170590929009>
- Kolarik, W. J., & Lu, S. S. (2001). Real-time performance reliability prediction. *IEEE Transactions on Reliability*, 50(4), 353–357.
<https://doi.org/10.1109/24.983393>
- Lu, S., Lu, H., & Kolarik, W. J. (2001). Multivariate performance reliability prediction in real-time, 72, 39–45.
- Manwell, J., & Mc Gowan, J. (2009). *WIND ENERGY WIND ENERGY* (This editi).
- Márquez, F. P. G., & Pedregal, D. J. (2007). Applied RCM2 algorithms based on statistical methods. *International Journal of Automation and Computing*, 4(2), 109–116. <https://doi.org/10.1007/s11633-007-0109-1>

- Ogata, K. (2004). *System Dynamics. Books, Fourth Edition University of Minnesota*, 777. <https://doi.org/10.1177/0583102480012001063.3.CO;2-C>
- P. O'Connor, A. K. (2012). *Practical Reliability Engineering* (the 5th ed). West Sussex: John Wiley & Sons.
- Pulkkinen, U., Simola, K., & Myo, E. (2006). Application of stochastic filtering for lifetime prediction, *91*, 200–208. <https://doi.org/10.1016/j.ress.2005.01.002>
- Rob J. Hyndman, Anne B. Koehler, J. K. O. and R. D. S. (2008). *Forecasting with Exponential Smoothing*. Springer.
- Sami, M., & Patton, R. J. (2012). *An FTC Approach to Wind Turbine Power Maximisation via T-S Fuzzy Modelling and Control. IFAC Proceedings Volumes* (Vol. 45). IFAC. <https://doi.org/10.3182/20120829-3-MX-2028.00131>
- Shipurkar, U., Ma, K., Polinder, H., Blaabjerg, F., & Ferreira, J. A. (2015). A review of failure mechanisms in wind turbine generator systems. *2015 17th European Conference on Power Electronics and Applications (EPE'15 ECCE-Europe)*, 1–10. <https://doi.org/10.1109/EPE.2015.7311669>
- Sloth, C., Esbensen, T., & Stoustrup, J. (2011). Aalborg Universitet Robust and Fault-Tolerant Linear Parameter-Varying Control of Wind Turbines Publication date : Mechatronics Robust and fault-tolerant linear parameter-varying control of wind turbines. *Mechatronics*, *21*(4), 645–659. <https://doi.org/10.1016/j.mechatronics.2011.02.001>
- Stephen A, D. (2001). *Instrumentation and Measurement*. Wiley.
- Taylor, P., Kim, Y. O. N. S., & Kolarik, W. J. (2007). Real-time conditional reliability prediction from on-line tool performance data, (December 2014), 37–41. <https://doi.org/10.1080/00207549208948125>
- Usher, J. S., Kamal, A. H., & Syed, W. H. (1998). Cost optimal preventive maintenance and replacement scheduling. *IIE Transactions (Institute of Industrial Engineers)*, *30*(12), 1121–1128. <https://doi.org/10.1080/07408179808966568>
- Xu, Z., Ji, Y., & Zhou, D. (2008). Real-time Reliability Prediction for a Dynamic System Based on the Hidden Degradation Process Identification, *57*(2), 230–

242.

- Xu, Z., Ji, Y., & Zhou, D. (2009). A new real-time reliability prediction method for dynamic systems based on on-line fault prediction. *IEEE Transactions on Reliability*, 58(3), 523–538. <https://doi.org/10.1109/TR.2009.2026785>
- Xu, Z., & Zhou, D. H. (2006). *a Degradation Measurements Based Real-Time Reliability Prediction Method*. *IFAC Proceedings Volumes* (Vol. 39). IFAC. <https://doi.org/10.3182/20060829-4-CN-2909.00158>

LAMPIRAN A

M-FILE SIMULASI PADA SITEM TURBIN ANGIN

```

clear all

clc

% Wind turbine data
Pnom = 400/0.9;
Pmec1 = 400;
Pmec=Pmec1*1;
wind0 = 11; % (m/s)
wind_speed_CpMax = 11; % (m/s)
Ts = 1e-1;
c1 = 0.6450;
c2 = 116;
c3 = .4;
c4 = 5;
c5 = 21;
c6 = .00912; % Set Pmin in stall condition (does not vary with
wind_speed_Pmax)
c7 = .08;
c8 = .035;
theta=0;
CpMax=.5;
lambda_CpMax=9.9495;

c1_c8=[c1 c2 c3 c4 c5 c6 c7 c8];

% Cp = c1*(c6*lambda + (-c4 - c3*(2.5 + theta) + c2*(1/(lambda +
...
% c7*(2.5 + theta)) - c8/(1 + (2.5 +
theta)^3)))/exp(c5*(1/(lambda + ...
% c7*(2.5 + theta)) - c8/(1 + (2.5 + theta)^3)))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Operation at Cp max
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        rated_omegar=1.2;
        omegar = rated_omegar;

% Enforce boundaries on wind speed
% A warning could be displayed
if wind_speed_CpMax < 6
        wind_speed_CpMax=6;

```

```

        disp('Warning: Wind speed at nominal speed and at Cp
max has been set to 6 m/s')
    end
    if wind_speed_CpMax > 30
        wind_speed_CpMax=30;
        disp('Warning: Wind speed at nominal speed and at Cp
max has been set to 30 m/s')
    end

    K1=lambda_CpMax/omegar*wind_speed_CpMax;

% Calculation of K2 needed to get P equal to the requested value
at
% Cp max for the requested wind speed

    Prated=Pmec1*1;
    P_rated_omegar_theta_zero=0.75;

    K2 =
P_rated_omegar_theta_zero*Prated/(wind_speed_CpMax^3*CpMax);

    K1_K2=[K1 K2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End of calculation for operation at Cp max
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Drive train data
H_WT = 4.32; % (s)
Ksh = 80.27; % (pu of nominal mechanical torque/rad)
F_WT = 0;
D_mutual = 1.5; % (pu of nominal mechanical torque/pu dw)
w_wt0 = 1.2; % (pu of nominal speed)
torque0 = 0.83; % (pu of nominal mechanical torque)

% Reference rotor speed (wref) data
Speed_max=1.2; % (pu)
Speed_min=0.5; % (pu)
T_speed = 5; %s

% Speed regulator data
Kp_speed = 3;
Ki_speed = 0.1;
Max_speed_reg=1; %Maximum output value for PI speed regulator (pu)

% Pitch control data
Kp_pitch = 0.1;
Ki_pitch = 0.5;
pitch_max = 27; % (deg)
pitch_rate = 10; % (deg/s)
Pitch_time_constant=0.01; % (s)
wn = 1;

```



```

zeta = 1;

% Pitch compensation data
Kp_compensation = 3;
Ki_compensation = 30;

% Generator data
H = 0.685; % Inertia constant (s)
F = 0.01; % Friction factor (pu)
wmo = 1.2; % (pu of nominal speed)
init = -0.2;
taug = 0.1;
ng = 1;

% pitch = 0;
% beta0 = -0.9;
% R = K1;
% dTadB = 0;
% seperlamdai = 1/(lambda_CpMax + c7*(beta0 + pitch)) - c8/(1 +
(beta0 + pitch)^3);
% dCpdlamda = c1*c6 + c1*c2*exp(-c5*seperlamdai)-c5*c1*(-c4 -
c3*(beta0+pitch) + c2*seperlamdai)*exp(-c5*seperlamdai);
% dTadwr = K2*R*wind_speed_CpMax^2*dCpdlamda-
K2/w_wt0*wind_speed_CpMax^3*CpMax;
% dTadVw = -R*K2*w_wt0*wind_speed_CpMax*dCpdlamda +
3*K2*wind_speed_CpMax^2*CpMax;

Ac = [-1/taug 0 0 0;
      0 -(D_mutual)/(2*H_WT) D_mutual/(2*H_WT) -Ksh/(2*H_WT);
      -1/(2*H) (D_mutual)/ng/(2*H) -(D_mutual+F)/ng^2/(2*H)
      Ksh/ng/(2*H);
      0 1 -1/ng 0];

Bc = [0 1/taug;
      1/(2*H_WT) 0;
      0 0;
      0 0];

C = [1 0 0 0; 0 1 0 0];
D = [0 0; 0 0];
sys = ss (Ac,Bc,C,D);
T = 1e-3;
[sysd,G]=c2d(sys,T,'zoh')
A = sysd.a
B = sysd.b

n = length(A);
p = size(C,1);
m = size(B,2);

gg1 = [0 1];
[Add,Dd]=c2d(A,eye(n),T);
Dw = [0; 1]; %[1; 1];
Fs = Dw;

```

```

dw = size(Dw,2);
dd = size(Dd,2);
fs = size(Fs,2);

Abar = 10*eye(p);
Aa = [A zeros(n,p); T*Abar*C eye(p)-T*Abar];
Ba = [B; zeros(p,m)];
Ea = [zeros(n,fs); T*Abar*Fs];
Da = [Dd ;zeros(p,dd)];
Ca = [zeros(p,n) eye(p)];
Bo = [Da zeros(n+p,fs); zeros(fs,dd) eye(fs)];
% Atild = A;
% Btild = B;
% Ctild = C;
Atild = [Aa Ea; zeros(fs,n+p) eye(fs)];
Btild = [Ba; zeros(fs,m)];
Ctild = [Ca zeros(p,fs)];

p1 = 0.92;
p2 = 0.91;
p3 = 0.938763;
p4 = 0.94;
p5 = 0.919;
p6 = 0.9911;
p7 = 0.99911;
Ktild = place(Atild',Ctild',[p1 p2 p3 p4 p5 p6 p7])

setlmis([])
P = lmivar(1,[n+p+fs 1]);
G = lmivar(2,[n+p+fs p]);
gamma = lmivar(1,[1 0]);

lmiterm([1 1 1 P],1,Atild,'s')
lmiterm([1 1 1 G],-1,Ctild,'s')
lmiterm([1 1 1 0],1)
lmiterm([1 1 2 P],1,Bo)
lmiterm([1 2 2 gamma],-1,1)

lmiterm([-2 1 1 P],1,1)

lmiterm([3 1 1 P],-1,1)
lmiterm([3 1 2 P],1,Atild)
lmiterm([3 1 2 P],0,1)
lmiterm([3 1 2 G],-1,Ctild)
lmiterm([3 2 2 P],-(0.4)^2,1)

LMIs = getlmis;

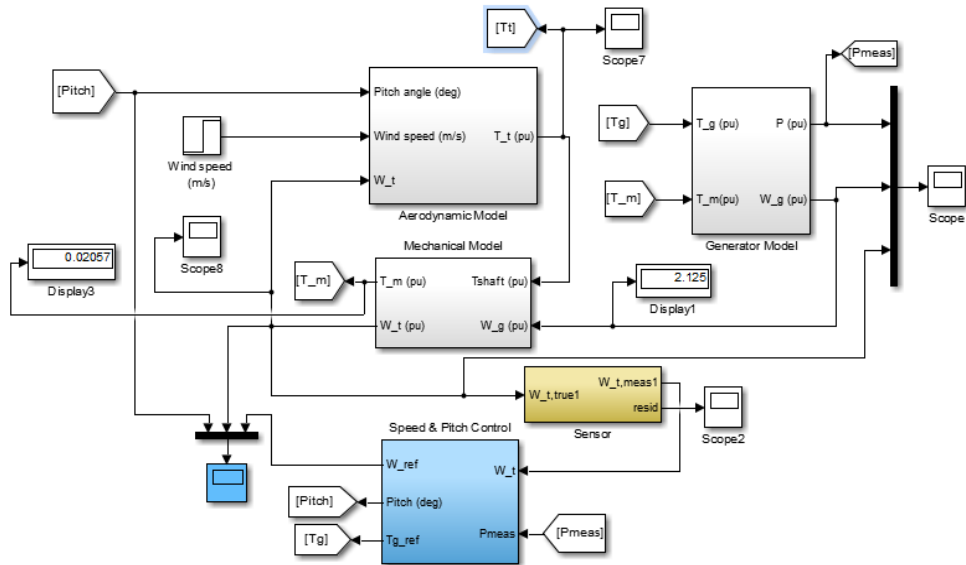
a = 0.01;
b = 0.09;
Tp = 10;

```

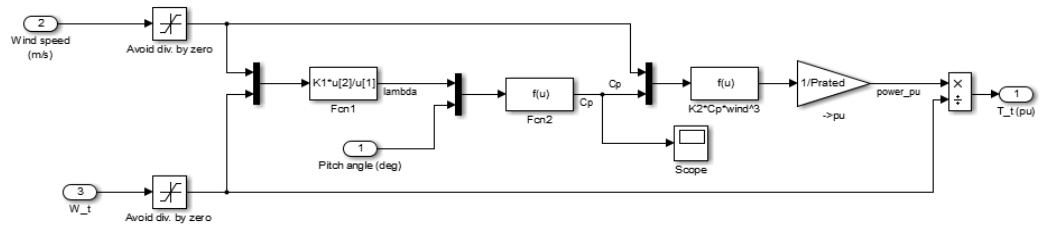
LAMPIRAN B

SIMULINK SISTEM TURBIN ANGIN

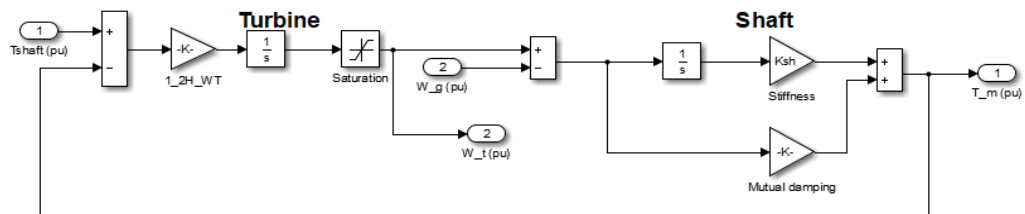
B.1 Sistem Turbin Angin



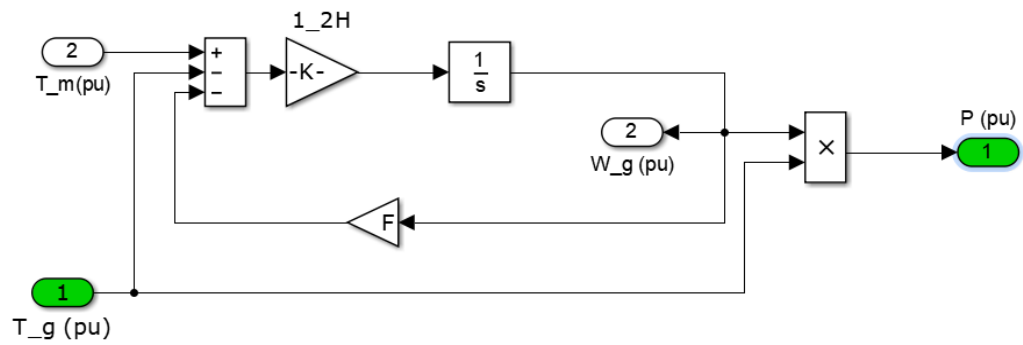
B.2 Model Aerodinamik dari Sistem Turbin Angin



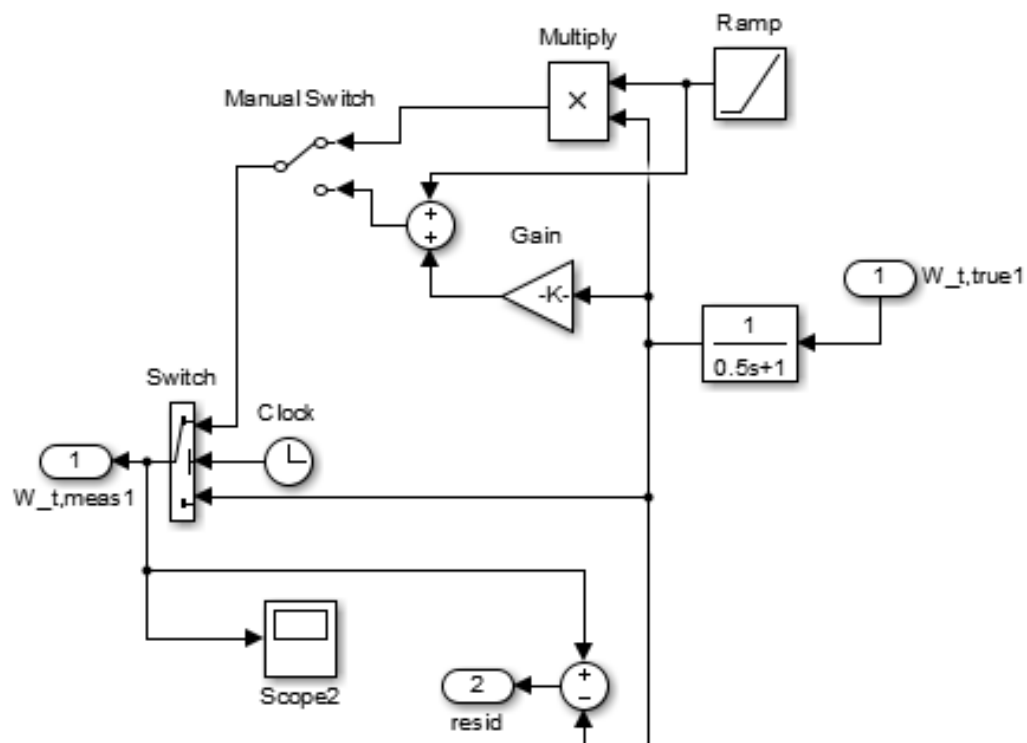
B.3 Model Mekanik dari Sistem Turbin Angin



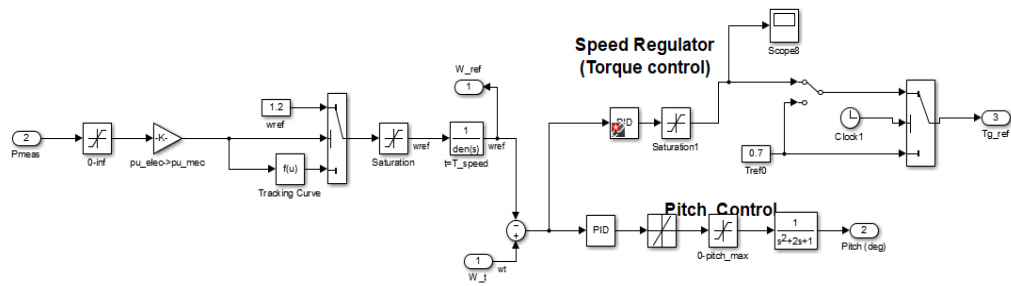
B.4 Model Generator dari Sistem Turbin Angin



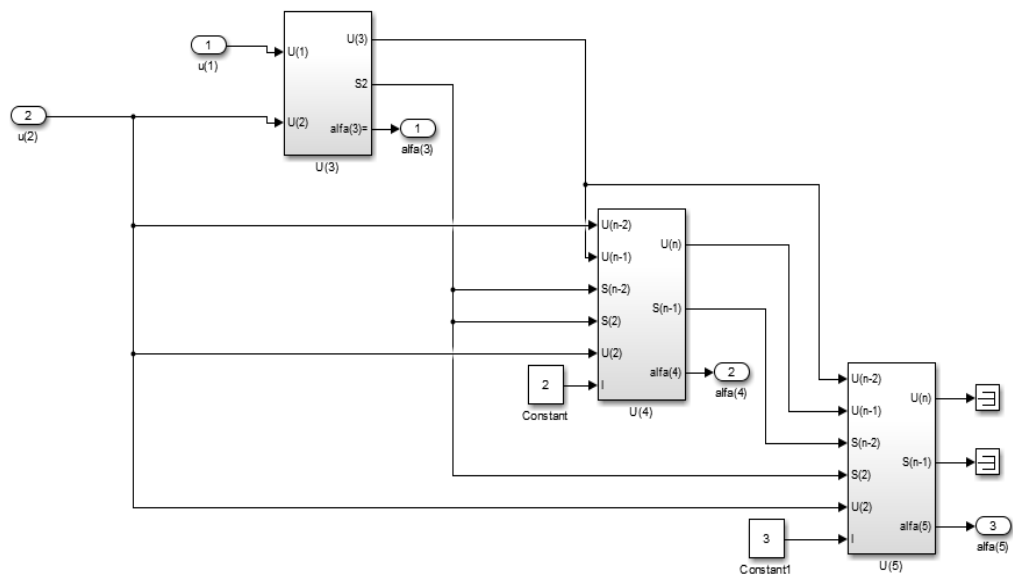
B.5 Model Sensor



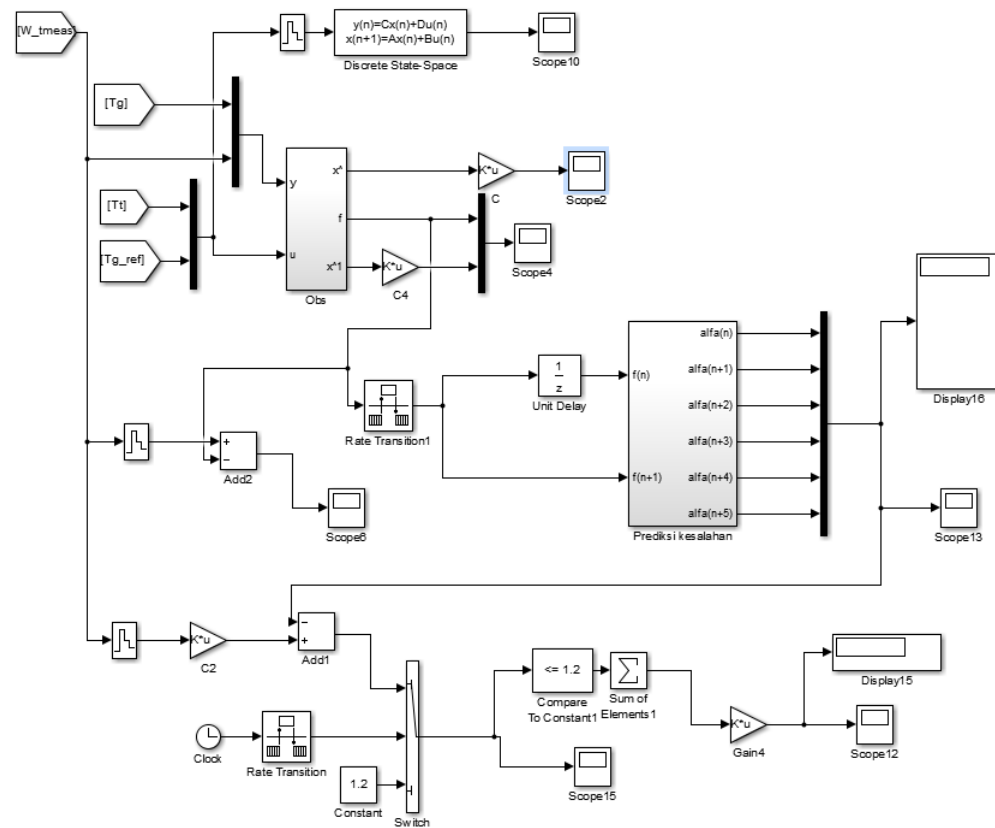
B.6 Sistem Pengendalian Kecepatan Turbin Angin



B.7 Model Prediksi Kesalahan Sensor ($\hat{f}(k + 5)$)



B.8 Observer, Prediksi Kesalahan Sensor dan Prediksi Keandalan *Real-time*



LAMPIRAN C

<i>Time Interval</i> (detik)	Waktu (detik)	Keandalan (Lima prediksi)	Waktu (detik)	Keandalan (Sepuluh prediksi)
1	0-1855	1	0-1850	1
	1855-1856	0.8	1850-1851	0.9
	1856-1857	0.6	1851-1852	0.8
	1857-1858	0.4	1852-1853	0.7
	1858-1859	0.2	1853-1854	0.6
	1859	0	1854-1855	0.5
			1855-1856	0.4
			1856-1857	0.3
			1857-1858	0.2
			1858-1859	0.1
			1859	0
2	0-1846	1	0-1824	1
	1846-1850	0.8	1824-1828	0.9
	1850-1854	0.6	1828-1832	0.8
	1854-1858	0.4	1832-1838	0.7
	1858-1860	0.2	1838-1842	0.6
	1860	0	1842-1846	0.5
			1846-1850	0.4
			1850-1854	0.3
			1854-1858	0.2
			1858-1860	0.1
			1860	0
3	0-1830	1	0-1761	1
	1830-1830	0.8	1761-1779	0.9
	1839-1851	0.6	1779-1794	0.8
	1851-1860	0.4	1794-1806	0.7
	1860-1863	0.2	1806-1818	0.6
	1863	0	1818-1830	0.5
			1830-1839	0.4
			1839-1851	0.3
			1851-1860	0.2
			1860-1863	0.1
			1863	0
4	0-1804	1	0-804	1

<i>Time Interval</i> (detik)	Waktu (detik)	Keandalan (Lima prediksi)	Waktu (detik)	Keandalan (Sepuluh prediksi)
	1804-1824	0.8	804-808	0.9
	1824-1844	0.6	808-816	0.8
	1844-1860	0.4	816-888	0.9
	1860-1864	0.2	888-1356	0.8
	1864	0	1356-1528	0.9
			1528-1692	0.8
			1692-1744	0.7
			1744-1776	0.6
			1776-1804	0.5
			1804-1824	0.4
			1824-1844	0.3
			1844-1860	0.2
			1860-1864	0.1
			1864	0
5	0-1755	1	0-805	1
	1755-1800	0.8	805-1670	0.6
	1800-1835	0.6	1670-1755	0.5
	1835-1860	0.4	1775-1800	0.4
	1860-1865	0.2	1800-1835	0.3
	1865	0	1835-1860	0.2
			1860-1865	0.1
			1865	0
6	0-810	1	0-804	1
	810-816	0.8	804-810	0.5
	816-1602	1	810-816	0.4
	1602-1758	0.8	816-1602	0.5
	1758-1818	0.6	1602-1758	0.4
	1818-1860	0.4	1758-1818	0.3
	1860-1866	0.2	1818-1860	0.2
	1866	0	1860-1866	0.1
			1866	0
7	0-805	1	0-805	1
	805-1680	0.8	805-1680	0.4
	1680-1799	0.6	1680-1799	0.3
	1799-1862	0.4	1799-1862	0.2
	1862-1869	0.2	1862-1869	0.1
	1869	0	1869	0
8	0-808	1	0-808	1

<i>Time Interval (detik)</i>	Waktu (detik)	Keandalan (Lima prediksi)	Waktu (detik)	Keandalan (Sepuluh prediksi)
	808-1776 1776-1864 1864-1872 1872	0.6 0.4 0.2 0	808-1776 1776-1864 1864-1872 1872	0.3 0.2 0.1 0
9	0-801 801-1737 1737-1863 1863-1872 1872	1 0.6 0.4 0.2 0	0-801 801-1737 1737-1863 1863-1872 1872	1 0.3 0.2 0.1 0
10	0-810 810-1660 1660-1860 1860-1870 0	1 0.6 0.4 0.2 0	0-810 810-1660 1660-1860 1860-1870 0	1 0.3 0.2 0.1 0
11	0-803 803-1859 1859-1870 1870	1 0.4 0.2 0	0-803 803-1859 1859-1870 1870	1 0.2 0.1 0
12	0-804 804-1860 1860-1872 1872	1 0.4 0.2 0	0-804 804-1860 1860-1872 1872	1 0.2 0.1 0
13	0-806 806-1859 1859-1872 1872	1 0.4 0.2 0	0-806 806-1859 1859-1872 1872	1 0.2 0.1 0
14	0-812 812-1862 1862-1876 1876	1 0.4 0.2 0	0-812 812-1862 1862-1876 1876	1 0.2 0.1 0
15	0-810 810-1860 1860-1875 1875	1 0.4 0.2 0	0-810 810-1860 1860-1875 1875	1 0.2 0.1 0
16	0-816 816-1872 1872-1888 1888	1 0.4 0.2 0	0-816 816-1872 1872-1888 1888	1 0.2 0.1 0

<i>Time Interval (detik)</i>	Waktu (detik)	Keandalan (Lima prediksi)	Waktu (detik)	Keandalan (Sepuluh prediksi)
17	0-816	1	0-816	1
	816-1870	0.4	816-1870	0.2
	1870-1887	0.2	1870-1887	0.1
	1887	0	1887	0
18	0-810	1	0-810	1
	810-1872	0.4	810-1872	0.2
	1872-1890	0.2	1872-1890	0.1
	1890	0	1890	0
19	0-817	1	0-817	1
	817-1862	0.4	817-1862	0.2
	1862-1881	0.2	1862-1881	0.1
	1881	0	1881	0
20	0-820	1	0-820	1
	820-1860	0.4	820-1860	0.2
	1860-1880	0.2	1860-1880	0.1
	1880	0	1880	0
21	0-819	1	0-819	1
	819-1869	0.4	819-1869	0.2
	1869-1890	0.2	1869-1890	0.1
	1890	0	1890	0
22	0-814	1	0-814	1
	814-1870	0.4	814-1870	0.2
	1870-1892	0.2	1870-1892	0.1
	1892	0	1892	0
23	0-805	1	0-805	1
	805-1863	0.4	805-1863	0.2
	1863-1886	0.2	1863-1886	0.1
	0	0	0	0
24	0-816	1	0-816	1
	816-1872	0.4	816-1872	0.2
	1872-1896	0.2	1872-1896	0.1
	1896	0	1896	0
25	0-825	1	0-825	1
	825-1875	0.4	825-1875	0.2
	1875-1900	0.2	1875-1900	0.1
	1900	0	1900	0
26	0-806	1	0-806	1
	806-1872	0.4	806-1872	0.2

<i>Time Interval</i> (detik)	Waktu (detik)	Keandalan (Lima prediksi)	Waktu (detik)	Keandalan (Sepuluh prediksi)
	1872-1898 1898	0.2 0	1872-1898 1898	0.1 0
27	0-810 810-1663 1663-1890 1890	1 0.4 0.2 0	0-810 810-1663 1663-1890 1890	1 0.2 0.1 0
28	0-812 812-1876 1876-1904 1904	1 0.4 0.2 0	0-812 812-1876 1876-1904 1904	1 0.2 0.1 0
29	0-812 812-1885 1885-1914 1914	1 0.4 0.2 0	0-812 812-1885 1885-1914 1914	1 0.2 0.1 0
30	0-810 810-1860 1860-1890 1890	1 0.4 0.2 0	0-810 810-1860 1860-1890 1890	1 0.2 0.1 0
31	0-806 806-1860 1860-1891 1891	1 0.4 0.2 0	0-806 806-1860 1860-1891 1891	1 0.2 0.1 0
32	0-832 832-1888 1888-1920 1920	1 0.4 0.2 0	0-832 832-1888 1888-1920 1920	1 0.2 0.1 0
33	0-825 825-1881 1881-1914 1914	1 0.4 0.2 0	0-825 825-1881 1881-1914 1914	1 0.2 0.1 0
34	0-816 816-1870 1870-1904 1904	1 0.4 0.2 0	0-816 816-1870 1870-1904 1904	1 0.2 0.1 0
35	0-805 805-1890 1890-1925 1925	1 0.4 0.2 0	0-805 805-1890 1890-1925 1925	1 0.2 0.1 0

<i>Time Interval</i> (detik)	Waktu (detik)	Keandalan (Lima prediksi)	Waktu (detik)	Keandalan (Sepuluh prediksi)
36	0-828	1	0-828	1
	828-1872	0.4	828-1872	0.2
	1872-1908	0.2	1872-1908	0.1
	1908	0	1908	0
37	0-814	1	0-814	1
	814-1887	0.4	814-1887	0.2
	1887-1924	0.2	1887-1924	0.1
	1924	0	1924	0
38	0-836	1	0-836	1
	836-1862	0.4	836-1862	0.2
	1862-1900	0.2	1862-1900	0.1
	1900	0	1900	0
39	0-819	1	0-819	1
	819-1872	0.4	819-1872	0.2
	1872-1911	0.2	1872-1911	0.1
	1911	0	1911	0
40	0-840	1	0-840	1
	840-1880	0.4	840-1880	0.2
	1880-1920	0.2	1880-1920	0.1
	1920	0	1920	0
41	0-820	1	0-820	1
	820-1886	0.4	820-1886	0.2
	1886-1927	0.2	1886-1927	0.1
	1927	0	1927	0
42	0-840	1	0-840	1
	840-1890	0.4	840-1890	0.2
	1890-1932	0.2	1890-1932	0.1
	1932	0	1932	0
43	0-817	1	0-817	1
	817-1892	0.4	817-1892	0.2
	1892-1935	0.2	1892-1935	0.1
	1935	0	1935	0
44	0-836	1	0-836	1
	836-1892	0.4	836-1892	0.2
	1892-1936	0.2	1892-1936	0.1
	1936	0	1936	0
45	0-810	1	0-810	1
	810-1890	0.4	810-1890	0.2

<i>Time Interval</i> (detik)	Waktu (detik)	Keandalan (Lima prediksi)	Waktu (detik)	Keandalan (Sepuluh prediksi)
	1890-1935 1935	0.2 0	1890-1935 1935	0.1 0
46	0-828 828-1886 1886-1932 1932	1 0.4 0.2 0	0-828 828-1886 1886-1932 1932	1 0.2 0.1 0
47	0-846 846-1880 1880-1972 1972	1 0.4 0.2 0	0-846 846-1880 1880-1972 1972	1 0.2 0.1 0
48	0-816 816-1872 1872-1920 1920	1 0.4 0.2 0	0-816 816-1872 1872-1920 1920	1 0.2 0.1 0
49	0-833 833-1862 1862-1911 1911	1 0.4 0.2 0	0-833 833-1862 1862-1911 1911	1 0.2 0.1 0
50	0-850 850-1900 1900-1950 1950	1 0.4 0.2 0	0-850 850-1900 1900-1950 1950	1 0.2 0.1 0

Halaman ini sengaja dikosongkan

BIODATA PENULIS



LILIK AYURANI merupakan nama lengkap penulis dengan nama panggilannya Lilik. Penulis dilahirkan di Sidoarjo 10 Agustus 1994 sebagai anak keempat dari empat bersaudara. Riwayat pendidikan penulis adalah SDN 1 Kesamben Wetan Driyorejo (2000-2006), SMP YPM 5 Driyorejo (2006-2009), SMAN 1 Driyorejo (2009-2012), Program Studi S1 Pendidikan Fisika UNESA (2012-2016).

Penulis diterima sebagai mahasiswa S2 Teknik Fisika ITS dengan bidang keahlian rekayasa instrumentasi industri pada tahun 2016 semester genap. Penulis fokus pada bidang rekayasa instrumentasi dan kontrol untuk turbin angin sebagai penelitian tesisnya. Penulis dapat dihubungi melalui email ayuranililik@gmail.com.

Halaman ini sengaja dikosongkan